

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РД  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РД  
«КОЛЛЕДЖ ЭКОНОМИКИ И ПРАВА»**

УТВЕРЖДАЮ

Зам. директора по учебной работе

А.Л. Джалилова А.Л. Джалилова

подпись

ФИО

« 2 » 09 2023 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО ДИСЦИПЛИНЕ**

**ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

**специальность 09.02.07 «Информационные системы и программирование»**

Дербент 2023

Организация – разработчик:

Государственное бюджетное профессиональное образовательное учреждение РД «Колледж экономики и права»

Разработчики:

Магомедова А.Н., преподаватель экономических дисциплин ГБПОУ РД «КЭиП»

Одобрена на заседании цикловой (предметной) комиссии профессиональных дисциплин и модулей по специальности 09.02.07 «Информационные системы и программирование»

Протокол № 1 от « 28 » 08 2023 г.

Председатель комиссии ГТДС / Гюльмагомедов Т.Х. /

## СОДЕРЖАНИЕ

<b>1. ОБЩИЕ ПОЛОЖЕНИЯ.....</b>	<b>4</b>
<b>2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, .....</b>	<b>5</b>
<b>ПОДЛЕЖАЩИЕ ПРОВЕРКЕ .....</b>	<b>5</b>
<b>3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....</b>	<b>7</b>
<b>4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ .....</b>	<b>9</b>
<b>ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ.....</b>	<b>9</b>
<b>4.1. Типовые задания для оценки освоения МДК.01.01 Разработка программных модулей.....</b>	<b>9</b>
<b>4.2. Типовые задания для оценки освоения МДК.01.02 Поддержка и тестирование программных модулей .....</b>	<b>20</b>
<b>4.3. Типовые задания для оценки освоения МДК.01.03 Разработка .....</b>	<b>23</b>
<b>мобильных приложений.....</b>	<b>23</b>
<b>4.4. Типовые задания для оценки освоения МДК.01.04 Системное .....</b>	<b>107</b>
<b>программирование.....</b>	<b>107</b>
<b>4.5. Типовые задания для экзамена квалификационного по ПМ.01 .....</b>	<b>18 ..... 110</b>
<b>5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ.....</b>	<b>111</b>
<b>ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....</b>	<b>26</b>

# ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (ФОС) разработан с целью установления соответствия образовательных достижений студентов требованиям программы подготовки специалистов среднего звена по профессиональному модулю ПМ.01 Разработка модулей программного обеспечения для компьютерных систем.

ФОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации.

ФОС текущего контроля используется для оперативного и регулярного управления учебной деятельностью студентов.

ФОС промежуточной аттестации студентов по профессиональному модулю предназначен для оценки степени достижения запланированных результатов обучения по завершению изучения междисциплинарных курсов профессионального модуля, экзамена (квалификационного) по завершению изучения профессионального модуля в целом.

Фонд оценочных средств разработан на основании:

1. ФГОС СПО 09.02.07 Информационные системы и программирование
2. Примерной основной образовательной программы по специальности 09.02.07 Информационные системы и программирование
3. Рабочей программы профессионального модуля ПМ.02 Осуществление интеграции программных модулей.
4. Учебного плана по специальности 09.02.07 Информационные системы и программирование
5. Положения о формах, периодичности и порядке текущего контроля успеваемости и промежуточной аттестации обучающихся
6. Положения о квалификационном экзамене
7. Шаблона фонда оценочных средств профессионального модуля

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом в рамках освоения профессионального модуля *ПМ.01 Разработка модулей программного обеспечения для компьютерных систем* является овладение студентами вида профессиональной деятельности *Разработка модулей программного обеспечения для компьютерных систем*, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 6	Проявлять гражданскопатриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках.

В результате освоения профессионального модуля студент должен: **иметь**

**практический опыт:**

ПО1 в разработке кода программного продукта на основе готовой спецификации на уровне модуля;

ПО2 в использовании инструментальных средств на этапе отладки программного продукта;

ПО3 в проведении тестирования программного модуля по определенному сценарию; ПО4 в разработке мобильных приложений. **уметь:**

У1 осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

У2 создавать программу по разработанному алгоритму как отдельный модуль;

У3 выполнять отладку и тестирование программы на уровне модуля;

У4 осуществлять разработку кода программного модуля на современных языках программирования;

У5 уметь выполнять оптимизацию и рефакторинг программного кода;

У6 оформлять документацию на программные средства. **знать:**

З1 основные этапы разработки программного обеспечения;

З2 основные принципы технологии структурного и объектно-ориентированного программирования;

З3 способы оптимизации и приемы рефакторинга;

З4 основные принципы отладки и тестирования программных продуктов.

### 3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Оценивание уровней сформированности профессиональных и общих компетенций проводится в рамках текущего и промежуточного контроля.

В результате освоения профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем студенты демонстрируют три уровня сформированности профессиональных компетенций: пороговый, базовый и повышенный.

Для каждого конкретного этапа формирования компетенции определены категории «знать», «уметь», «практический опыт», в которые вкладывается следующий смысл:

«приобрести практический опыт» – решать усложненные задачи на основе приобретенных умений и навыков, с их применением в профессиональных деятельности;

«уметь» – решать типичные задачи на основе воспроизведения стандартных алгоритмов решения;

«знать» - воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты.

Наименование Разделов, МДК, тем	Результаты обучения - коды ПК, ОК	Приобретенный практический опыт, освоенные умения и знания	Уровень освоения ПК, ОК	Формы и методы контроля и оценки
1	2	3	4	5
<i>Раздел 1. Разработка программных модулей</i>				
МДК.01.01 Разработка программных модулей	ПК 1.1 ПК 1.2 ПК 1.4 ПК 1.5 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> У <sub>5</sub> У <sub>6</sub> З <sub>1</sub> З <sub>2</sub> З <sub>3</sub>	2,3	<input type="checkbox"/> оценка на практических занятиях; <input type="checkbox"/> оценка защиты лабораторных работ; <input type="checkbox"/> оценка выполнения индивидуального задания; <input type="checkbox"/> экзамен по МДК 01.01
Тема 1.1.1 Жизненный цикл ПО	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>6</sub> З <sub>1</sub>	2,3	
Тема 1.1.2 Структурное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.3 Объектноориентированное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.4 Паттерны проектирования	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.5 Событийноуправляемое программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub>	2,3	
Тема 1.1.6 Оптимизация и рефакторинг кода	ПК 1.4 ПК 1.5	ПО <sub>2</sub> ПО <sub>3</sub> У <sub>4</sub> У <sub>5</sub> З <sub>3</sub>	2,3	

	ОК 1-ОК 9			
1.1.7 Разработка пользовательского интерфейса	ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>2</sub>	2,3	
Тема 1.1.8 Основы ADO.Net	ПК 1.1 ПК 1.2 ОК 1-ОК 9	ПО <sub>1</sub> У <sub>1</sub> У <sub>2</sub> У <sub>4</sub> З <sub>2</sub>	2,3	
<i>Раздел 2. Поддержка и тестирование программных модулей</i>				
МДК.01.02 Поддержка и тестирование программных модулей	ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>2</sub> ПО <sub>3</sub> У <sub>3</sub> У <sub>6</sub> З <sub>1</sub> З <sub>4</sub>	2,3	<input type="checkbox"/> оценка на практических занятиях; оценка <input type="checkbox"/> защиты лабораторных работ; оценка <input type="checkbox"/> выполнения индивидуального задания; экзамен по МДК <input type="checkbox"/> 01.02
Тема 1.2.1 Отладка и тестирование программного обеспечения	ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>2</sub> ПО <sub>3</sub> У <sub>3</sub> З <sub>1</sub> З <sub>4</sub>	2,3	
1.2.2 Документирование	ПК 1.2 ОК 1-ОК 9	У <sub>6</sub> З <sub>1</sub> З <sub>4</sub>	2,3	
<i>Раздел 3. Разработка мобильных приложений</i>				
МДК.01.03 Разработка мобильных приложений	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> ПО <sub>4</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> оценка на практических занятиях; оценка <input type="checkbox"/> защиты лабораторных работ; оценка <input type="checkbox"/> выполнения индивидуального задания; экзамен по МДК 01.03
Тема 1.3.1 Основные платформы и языки разработки мобильных приложений	ПК 1.6 ОК 1-ОК 9	ПО <sub>4</sub> У <sub>4</sub> З <sub>2</sub>	2,3	
Тема 1.3.2 Создание и тестирование модулей для мобильных приложений	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ПК 1.6 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> ПО <sub>4</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	
<i>Раздел 4. Системное программирование</i>				
МДК.01.04 Системное программирование	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	<input type="checkbox"/> оценка на практических занятиях; оценка <input type="checkbox"/> защиты лабораторных работ; оценка

Тема 1.4.1 Программирование на языке низкого уровня	ПК 1.1 ПК 1.2 ПК 1.3 ПК 1.4 ОК 1-ОК 9	ПО <sub>1</sub> ПО <sub>2</sub> ПО <sub>3</sub> У <sub>1</sub> У <sub>2</sub> У <sub>3</sub> У <sub>4</sub> З <sub>1</sub> З <sub>2</sub> З <sub>4</sub>	2,3	<input type="checkbox"/> выполнения индивидуального задания; экзамен по МДК <input type="checkbox"/> 01.04
---	---	--	-----	--

#### 4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

##### 4.1. Типовые задания для оценки освоения МДК.01.01 Разработка программных модулей

##### Примеры тестовых заданий для текущего контроля

**1. Максимально возможная интеграция программных продуктов с другими программами, обеспечение обмена данными в общих форматах представления (экспорт/импорт баз данных, внедрение или связывание объектов обработки и др.) означает их...**

- a. модифицируемость
- b. эффективность
- c. мобильность
- d. коммуникативность

**2. Наличие дружественного интерфейса, контекстно-зависимой подсказки, хорошей документации является таким показателем качества программного продукта, как ...**

- a. коммуникативность
- b. эффективность
- c. мобильность
- d. учет человеческого фактора

**3. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:**

- постановки задачи,
- выбора критериев эффективности,
- проведения предварительных научно-исследовательских работ,
- разработки технического задания, определяет содержание этапа модели ЖЦ ...

- a. Эскизный проект
- b. Технический проект
- c. Техническое задание
- d. Рабочий проект

**4. Наличие обратных связей между этапами жизненного цикла, определяющих возможность проведения проверок и корректировок проектируемого ПС на каждой стадии разработки, является основной особенностью ...**

- a. каскадной модели жизненного цикла
- b. итерационной модели жизненного цикла
- c. спиральной модели жизненного цикла ПО 18
- d. модели жизненного цикла UML

**5. Функционирование программного продукта в соответствии с техническим заданием соответствует характеристике эксплуатационных требований ...**

- a. правильность

- b. универсальность
- c. надежность
- d. адаптируемость

**6. Обеспечение правильности результатов при наличии различного рода сбоях соответствует характеристике эксплуатационных требований ...**

- a. правильность
- b. универсальность
- c. надежность
- d. защищенность

**7. Обеспечение погрешности результатов не выше заданной соответствует характеристике эксплуатационных требований...**

- a. правильность
- b. проверяемость
- c. точность результатов
- d. защищенность

**8. Функциональная диаграмма имеет аббревиатуру ...**

- a. ERD
- b. SADT
- c. DFD
- d. STD

**9. Для описания основных (стандартных) типов данных на языке C++ определены следующие ключевые слова...**

- a. int, char, bool, double, void
- b. int, char, float, double, bool
- c. int, wchar\_t, float, double, char

**10. При обращении к функции в соответствии с требованиями хорошего стиля программирования рекомендуется (но не обязательно), чтобы ...**

- a. имена формальных и фактических параметров были различными
- b. количество формальных параметров соответствовало количеству фактических параметров
- c. типы формальных параметров соответствовали типам фактических параметров
- d. порядок перечисления фактических параметров соответствовал порядку перечисления формальных параметров

**11. В объектно-ориентированной среде программирования обработчики любых событий для компонентов размещаются в окне ...**

- a. Формы
- b. Инспектора объектов
- c. Кода
- d. Деревя объектов

**12. В объектно-ориентированной среде программирования к любому компоненту в программе можно обратиться по его свойству ...**

- a. Caption
- b. Text
- c. Align
- d. Name

**13. Абстрактным называется класс, который ...**

- a. не содержит функций
- b. не содержит переменных

- c. содержит хотя бы одну виртуальную функцию
- d. содержит хотя бы одну чисто виртуальную функцию

**14. Модульное тестирование предполагает ...**

- a. тестирование минимально возможного для тестирования компонента, например, отдельного класса или функции
- b. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- c. тестирование интегрированной системы на ее соответствие исходным требованиям
- d. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

**15. Интеграционное тестирование предполагает ...**

- a. тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
- b. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- c. тестирование интегрированной системы на ее соответствие исходным требованиям
- d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**16. Системное тестирование предполагает ...**

- a) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- b) тестирование интегрированной системы на ее соответствие исходным требованиям
- c) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
- d) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**17. Альфа-тестирование предполагает ...**

- a) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- b) тестирование интегрированной системы на ее соответствие исходным требованиям
- c) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
- d) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**18. Бета-тестирование предполагает ...**

- a) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- b) тестирование интегрированной системы на ее соответствие исходным требованиям
- c) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
- d) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**19. Видимость переменных и методов класса только в пределах данного модуля определяет спецификатора доступа ...**

- a. private
- b. protected
- c. public
- d. published

**20. Видимость переменных и методов класса в любом месте программы определяет спецификатора доступа ...**

- a. private

- b. protected
- c. public
- d. published

**21. Процесс, в ходе которого один объект может приобретать свойства другого, называется...**

- a. наследование
- b. инкапсуляция
- c. полиморфизм
- d. динамический полиморфизм

**22. На основе наследования и виртуальных функций реализуется принцип ...**

- a. наследования
- b. инкапсуляции
- c. статического полиморфизма
- d. динамического полиморфизма

**23. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - \*.**

```
int i, N = 20;
for(i = 0; i < N; i--)
    printf("*");
```

- a) int i, N = 40;for(i = 0; i < N; i--) printf("\*");
- b) int i, N = 20;for(i = 0; i < N; N--) printf("\*");
- c) int i, N = 20;for(i = 20; i < N; i--) printf("\*");
- d) int i, N = 20;for(i = 19; i < N; i--) printf("\*");

**24.Какой индекс имеет последний элемент массива A[6]?**

- a. int A[6];
- b. int A[7];
- c. int A[5];
- d. int A[0];

**25.Какими знаками заканчивается большинство строк кода в C++?**

- a): (двоеточие);
- b) . (точка);
- c) ; (точка с запятой);
- d), (запятая)

**26.Какой из перечисленных типов данных не является типом данных в C++?**

- a)real;
- b) int;
- c)double;
- d) float

**27.Какая из следующих записей - правильный комментарий в C++?**

- a)\*/ Комментарий \*/ ;
- b){комментарий};
- c)\*\* Комментарий \*\*;
- d)/\* комментарий \*/

**28. Что будет напечатано на экране, после выполнения этого кода?**

```
#include <iostream>
int foo(int x, int y)
{
    return x+y;
}
double foo(double x, double y)
{
    return x+y;
```

```

}
int main(int argc, char** argv)
{
    double (*ptr)(int, int);
    ptr = foo;
    std::cout << ptr(3,8) << std::endl;
    return 0;
}

```

- a. 3
- b. 11
- c. ошибка компиляции
- d. 8

**29. Что будет напечатано на экране, после выполнения этого кода?**

```

#include <iostream>
int foo(int y);
int foo(int x)
{
    return x+1;
}
int main(int argc, char** argv)
{
    int x = 3;
    int y = 6;
    std::cout << foo(x) << std::endl;
    return 0;
}

```

- a. 4
- b. ошибка компиляции
- c. 3
- d. 9

**30. Класс - это:**

- a) любой тип данных, определяемый пользователем
- b) тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- c) структура, для которой в программе имеются функции работы с нею

**31. Членами класса могут быть?**

- a) как переменные, так и функции, могут быть объявлены как private и как public
- b) только переменные, объявленные как private
- c) только функции, объявленные как private
- d) только переменные и функции, объявленные как private
- e) только переменные и функции, объявленные как public

**32. Что называется конструктором?**

- a) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- b) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- c) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- d) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

**33. Объект - это**

- a) переменная, содержащая указатель на класс
- b) экземпляр класса
- c) класс, который содержит в себе данные и методы их обработки

#### 34. Отметьте правильные утверждения

- a) конструкторы класса не наследуются
- b) конструкторов класса может быть несколько, их синтаксис определяется программистом
- c) конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- d) конструктор возвращает указатель на объект
- e) конструктор не возвращает значение

#### 35. Что называется наследованием?

- a) это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы
- b) это механизм переопределения методов базового класса
- c) это механизм, посредством которого производный класс получает все поля базового класса
- d) это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

#### 36. Выберите правильные утверждения:

- a) если элементы класса объявлены как `private`, то они доступны только наследникам класса, но не внешним функциям
- b) если элементы класса объявлены как `private`, то они недоступны ни наследникам класса, ни внешним функциям
- c) если элементы объявлены как `public`, то они доступны наследникам класса, но не внешним функциям
- d) если элементы объявлены как `public`, то они доступны и наследникам класса, и внешним функциям

#### 37. Возможность и способ обращения производного класса к элементам базового определяется

- a) ключами доступа: `private`, `public`, `protected` в теле производного класса
- b) только ключом доступа `protected` в заголовке объявления производного класса
- c) ключами доступа: `private`, `public`, `protected` в заголовке объявления производного класса
- d) ключами доступа: `private`, `public`, `protected` в теле базового класса

#### 38. Шаблон функции - это...

- a) определение функции, в которой типу обрабатываемых данных присвоено условное обозначение
- b) прототип функции, в котором вместо имен параметров указан условный тип
- c) определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
- d) определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

#### 39. Выберите правильные утверждения:

- a) по умолчанию члены класса имеют атрибут `private`
- b) по умолчанию члены класса имеют атрибут `public`;
- c) члены класса имеют доступ только к элементам `public`;
- d) элементы класса с атрибутом `private` доступны только членам класса

#### 40. Переопределение операций имеет вид:

- a) `имя_класса`, ключевое слово `operation`, символ операции
- b) `имя_класса`, ключевое слово `operator`, символ операции, в круглых скобках могут быть указаны аргументы
- c) `имя_класса`, ключевое слово `operator`, список аргументов
- d) `имя_класса`, два двоеточия, ключевое слово `operator`, символ операции

#### 41. Полиморфизм – это :

- a) средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов;
- b) средство, позволяющее в одном классе использовать методы с одинаковыми именами;
- c) средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий

d) средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

**42. Полиморфизм реализован через механизмы:**

- a) перегрузки функций, виртуальных функций, шаблонов
- b) перегрузки функций, наследования методов, шаблонов;
- c) наследования методов, виртуальных функций, шаблонов
- d) перегрузки функций, наследования, виртуальных функций.

**43. Без какой библиотеки не будет работать ввод/вывод данных?**

- a) math;
- b) random;
- c) iostream;
- d) string;

**44. какая команда используется для ввода/вывода данных в C++?**

- a) readln / Writeln ;
- b) cin/cout;
- c) alert/ prompt

**Вопросы для подготовки к экзамену/зачёту**

1. Трёхслойная клиент-серверная архитектура
2. Основные элементы HTML
3. Основные теги HTML
4. Форматирование текста в HTML. Символьные элементы
5. Гиперссылки в HTML
6. Фреймы в HTML
7. Таблицы в HTML. Атрибуты таблиц и ячеек
8. Таблицы в HTML. Объединение ячеек таблиц
9. Списки в HTML. Упорядоченные и неупорядоченные списки
10. Формы в HTML. Основные элементы форм. Атрибуты форм
11. Графические возможности HTML
12. Карты изображений в HTML
13. Каскадные таблицы стилей и их подключение к HTML-документу
14. Структура документа с PHP-скриптом. Включение PHP в HTML
15. Переменные и константы в PHP
16. Типы данных PHP. Преобразование типов
17. Операторы PHP. Приоритет операторов
18. Вывод контента в PHP.
19. Функции даты и времени в PHP
20. Строки в PHP. Функции работы со строками
21. Условный оператор в PHP
22. Оператор выбора в PHP
23. Циклы с заданным числом повторений в PHP
24. Циклы с предусловием и постусловием в PHP
25. Функции в PHP. Повторное использование кода
26. Массивы в PHP. Массивы с числовыми индексами и ассоциативные массивы
27. Массивы в PHP. Функции для работы с массивами

28. Обработка форм в PHP
29. Основные понятия JavaScript, структура программы, поддержка браузерами
30. Вывод результатов работы программы и ввод данных в JavaScript.
31. Переменные и типы данных JavaScript
32. Операторы JavaScript. Приоритет выполнения операторов
33. Типы данных JavaScript, преобразование типов данных
34. Массивы в JavaScript. Назначение массивов, объявление и инициализация массивов
35. Пользовательские функции JavaScript. Расположение функций внутри HTML-кода
36. Оператор ветвления в JavaScript
37. Оператор выбора JavaScript
38. Многократное выполнение блоков кода. Цикл с заданным числом повторений в JavaScript
39. Циклы с предусловием и постусловием в JavaScript
40. Встроенный класс Global в JavaScript. Основные функции и примеры использования
41. Встроенный класс Number в JavaScript. Работа с числами
42. Встроенный класс String в JavaScript. Обработка строк
43. Встроенный класс Array в JavaScript. Работа с массивами
44. Встроенный класс Math в JavaScript. Использование математических функций
45. Встроенный класс Date в JavaScript. Работа с датой и временем
46. События в JavaScript. Определение событий, обработчики событий. События мыши, пример использования
47. События в JavaScript. Определение событий, обработчики событий. События документа, пример использования
48. События в JavaScript. Определение событий, обработчики событий. События формы, пример использования.
49. События в JavaScript. Написание обработчиков событий, примеры
50. События в JavaScript. Объект event
51. Объектная модель документа DOM. Функции для работы с объектной моделью документа
52. Доступ к значениям элементов форм в JavaScript
53. Описание среды программирования на C++
54. Структура программы на языке C++
55. Понятие алгоритма. Свойства алгоритмов
56. Базовые структуры алгоритмов. Способы описания алгоритмов
57. Переменные и константы в языке C++
58. Типы данных в языке C++. Явное и неявное преобразование типов
59. Операции в языке C++
60. Математические функции в языке C++. Примеры использования математических функций

61. Условный оператор в языке C++. Назначение. Формат записи, блок-схема. Примеры
62. Оператор выбора в языке C++. Назначение. Формат записи, блок-схема. Примеры
63. Циклы с условием в языке C++. Назначение. Формат записи, блок-схема. Примеры
64. Циклы с заданным числом повторений в языке C++. Назначение. Формат записи, блок-схема. Примеры
65. Операторы передачи управления в языках высокого уровня.
66. Алгоритм нахождения суммы ряда с заданной точностью. Блок-схема алгоритма и реализация на языках высокого уровня
67. Нахождение корней уравнения методом половинного деления. Блок-схема алгоритма и реализация на языках высокого уровня
68. Нахождение корней уравнения методом хорд. Блок-схема алгоритма и реализация на языках высокого уровня
69. Вычисление определенного интеграла методом прямоугольников. Блок-схема алгоритма и реализация на языках высокого уровня
70. Вычисление определенного интеграла методом трапеций. Блок-схема алгоритма и реализация на языках высокого уровня
71. Одномерные массивы. Объявление и инициализация одномерных массивов
72. Случайные и псевдослучайные числа. Генерация псевдослучайных чисел в языках высокого уровня
73. Получение и изменение значений элементов массива. Перебор элементов массива
74. Матрицы. Объявление и инициализация матриц
75. Получение и изменение значений элементов матрицы. Перебор элементов матрицы
76. Нахождение минимального и максимального элемента массива. Блок-схема алгоритма и реализация в языках высокого уровня
77. Нахождение суммы элементов массива. Блок-схема алгоритма и реализация на языках высокого уровня
78. Сортировка массивов методом нахождения максимального элемента
79. Сортировка массивов методом «пузырька»
80. Указатели и операции над указателями. Выражения и арифметические действия с указателями
81. Взаимосвязь между указателями и массивами
82. Динамические массивы
83. Объявление и инициализация строк. Доступ к символам строки. Перебор символов строки
84. Основные функции для работы со строками
85. Пользовательские функции в языках высокого уровня. Объявление, определение и вызов функций
86. Пользовательские функции в языке языках высокого уровня. Расположение объявления и определения функций. Заголовочные файлы
87. Пользовательские функции в языках высокого уровня. Способы передачи параметров в функцию

88. Пользовательские функции в языках высокого уровня. Передача массивов в функции
89. Пользовательские функции в языках высокого уровня. Статические переменные
90. Пользовательские функции в языках высокого уровня. Рекурсия
91. Работа с файлами в языках высокого уровня. Открытие и закрытие файла
92. Работа с файлами в языках высокого уровня. Запись в файл и чтение из файла
93. Структуры. Объявление и инициализация структур
94. Структуры. Доступ к элементам структур
95. Структуры. Массивы структур
96. Распределение оперативной памяти для программ на языках высокого уровня. Динамическая память
97. Функции для работы с динамической памятью
98. Динамические структуры данных. Классификация динамических структур
99. Динамические структуры данных. Объявление динамических структур
100. Динамические структуры данных. Доступ к данным в динамических структурах
101. Динамические структуры данных. Работа с памятью при использовании динамических структур
102. Однонаправленные списки. Определение, основные операции с однонаправленным списком
103. Однонаправленные списки. Создание и просмотр однонаправленного списка
104. Однонаправленные списки. Вставка элемента в однонаправленный список
105. Однонаправленный список. Удаление элемента из однонаправленного списка
106. Однонаправленный список. Поиск элемента в однонаправленном списке
107. Стек. Определение, основные операции со стеком
108. Очередь. Определение, основные операции с очередью
109. Основные понятия объектно-ориентированного программирования
110. Классы и объекты. Объявление классов в языках высокого уровня
111. Классы и объекты. Объявление и определение атрибутов и методов
112. Классы и объекты. Конструкторы и деструкторы
113. Классы и объекты. Статические атрибуты и методы
114. Динамическое создание объектов. Указатели на объекты
115. Наследование. Множественное наследование
116. Виртуальные методы
117. Абстрактные методы и классы
118. Перегрузка операций
119. Платформа Microsoft .NET. Структура платформы
120. Платформа Microsoft .NET. Выполнение программы в .NET
121. Основные принципы объектно-ориентированного программирования
122. Среда Visual Studio .NET. Типы создаваемых проектов. Основные окна программы

## Примеры практических заданий

1. Создать форму для ввода массива и кнопкой «Перемешать». При нажатии на кнопку элементы массива перемешиваются
2. Палиндромом называют последовательность символов, которая читается как слева направо, так и справа налево. Создать web-форму для ввода строки и кнопкой, при нажатии на которую определяется, является ли введенная строка палиндромом
3. Создать web-форму с двумя полями для ввода чисел и выбором операции (+, -, \*, /).  
Вычислить соответствующее выражение и вывести результат в браузер
4. Найти наибольшее из трех чисел, введенных пользователем в форму
5. Написать программу нахождения синуса и косинуса угла, выбор действия осуществляется с помощью переключателя
6. Средствами PHP вывести время и дату на странице
7. Создать web-форму для ввода строки и кнопкой, при нажатии на которую строка переворачивается и выводится в браузер
8. Написать программу, которая для введенной строки подсчитывает сумму всех содержащихся в ней цифр
9. Создать класс Point, разработав следующие элементы класса: поля (x, y); конструкторы, позволяющие создать экземпляр класса с нулевыми координатами или с заданными координатами; методы, позволяющие вывести координаты точки на экран, рассчитать расстояние от начала координат до точки, переместить точку на плоскости на вектор (a, b); свойства для получения-установки координаты точки (доступное для чтения и записи), для умножения координаты точки на скаляр (доступное только для записи)
10. Создать класс Triangle, разработав следующие элементы класса: поля (a, b, c); конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; методы, позволяющие вывести длины сторон треугольника на экран, рассчитать периметр треугольника, рассчитать площадь треугольника; свойства для получения-установки длин сторон треугольника (доступное для чтения и записи), для определения, существует ли треугольник с данными длинами сторон (доступное только для чтения)
11. Создать класс Rectangle, разработав следующие элементы класса: поля (a, b); конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; методы, позволяющие вывести длины сторон прямоугольника на экран, рассчитать периметр прямоугольника, рассчитать площадь прямоугольника; свойства для получения-установки длин сторон прямоугольника (доступное для чтения и записи), для определения, является ли данный прямоугольник квадратом (доступное только для чтения)
12. Создать класс Round, разработав следующие элементы класса: поля (r); конструктор, позволяющий создать экземпляр класса с заданным радиусом; методы, позволяющие рассчитать длину окружности, рассчитать площадь круга; свойства для получения-установки радиуса круга (доступное для чтения и записи)
13. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран. Создать производные классы: Triangle (треугольник), Circle (круг) со своими методами вычисления площади и периметра. Создать массив n фигур и вывести полную информацию о фигурах на экран
14. Посчитать сумму элементов побочной диагонали квадратной матрицы

15. Найти среднее арифметическое положительных элементов одномерного массива. Количество элементов массива вводится пользователем, массив заполняется случайными числами
16. Посчитать в строке количество цифр. Строка вводится пользователем с клавиатуры
17. Создать форму Windows. Программа должна предоставлять возможность менять цвет фона формы
18. Создать Windows-приложение, демонстрирующее работу с модальными и немодальными окнами
19. Написать программу на языке C++, которая считает сумму элементов одномерного массива, не используя операцию индексирования [].
20. Написать программу на языках высокого уровня ,демонстрирующую способы передачи параметров в функции
21. Написать рекурсивную функцию для нахождения факториала числа
22. Написать программу на языках высокого уровня для подсчета суммы элементов одномерного массива. Размерность массива вводится пользователем с клавиатуры
23. Написать программу, демонстрирующую возможность создания односвязного списка и добавления элемента в список
24. Написать программу, демонстрирующую возможность создания стека и добавления элемента в любое место стека
25. Создать класс «Прямоугольник», который имеет два конструктора. Первый конструктор принимает в качестве параметров длину и ширину. Второй конструктор создает квадрат с заданной стороной
26. Написать программу, демонстрирующую работу с абстрактными классами
27. Создать функцию на языках высокого уровня, подсчитывающую сумму элементов массива. Массив передается как параметр в функцию
28. Написать программу для нахождения максимально элемента матрицы.
29. Написать программу для сортировки массива методом пузырька.
30. Написать программу для нахождения корней квадратного уравнения  $ax^2+bx+c=0$ . Коэффициенты a, b, c вводятся пользователем

#### **4.2. Типовые задания для оценки освоения МДК.01.02 Поддержка и тестирование программных модулей**

##### **1. Отметьте все правильные утверждения о массивах в языке C++**

- a) элементы массива могут быть разных типов;
- b) все элементы массива должны быть одного типа;
- c)элементы в памяти расположены рядом;
- d)элементы могут быть расположены в памяти по одному;
- e) элементы всегда нумеруются с нуля.

##### **2.Какой индекс имеет последний элемент массива A?**

`int A[6];`

##### **3.Какими знаками заканчивается большинство строк кода в C++?**

a): (двоеточие); b) . (точка); c) ; (точка с запятой); d), (запятая)

##### **4.Какой из перечисленных типов данных не является типом данных в C++?**

a)real; b) int; c)double; d) float

##### **5.Какая из следующих записей - правильный комментарий в C++?**

- a)\*/ Комментарии \*/; b){комментарий}; c)\*\* Комментарий \*\*;  
d)/\* комментарий \*/

**6.Оператор ввода cin может задавать ввод нескольких переменных в одной команде, используя следующий синтаксис:**

- a)cin » <переменная1> » <переменная2>...;  
b)cin » <переменная1>,<переменная2>,...;  
c)» <переменная1> » <переменная2> » endl »...;  
d)Все ответы верны

**7.Результат выполнения следующего фрагмента кода: cout << 22 / 5 \* 3;**

- a)13.2; b)другое; c)1; d)1.47; e)12.

**8. Класс - это:**

- a) любой тип данных, определяемый пользователем  
b) тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки  
c) структура, для которой в программе имеются функции работы с нею

**9.Членами класса могут быть?**

- a) как переменные, так и функции, могут быть объявлены как private и как public  
b) только переменные, объявленные как private  
c) только функции, объявленные как private  
d) только переменные и функции, объявленные как private  
e) только переменные и функции, объявленные как public

**10. Что называется конструктором?**

- a) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса  
b) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)  
c) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса  
d) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

**11.Объект - это**

- a) переменная, содержащая указатель на класс  
b) экземпляр класса  
c) класс, который содержит в себе данные и методы их обработки

**12. Отметьте правильные утверждения**

- a)конструкторы класса не наследуются  
b)конструкторов класса может быть несколько, их синтаксис определяется программистом  
c) конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций  
d) конструктор возвращает указатель на объект  
e) конструктор не возвращает значение

**13. Что называется наследованием?**

- a) это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы  
b)это механизм переопределения методов базового класса  
c)это механизм, посредством которого производный класс получает все поля базового класса  
d)это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

**14. Выберите правильные утверждения:**

- a)если элементы класса объявлены как private, то они доступны только наследникам класса, но не внешним функциям  
b) если элементы класса объявлены как private, то они недоступны ни наследникам класса, ни внешним функциям

с) если элементы объявлены как public, то они доступны наследникам класса, но не внешним функциям

д) если элементы объявлены как public, то они доступны и наследникам класса, и внешним функциям

**15. Возможность и способ обращения производного класса к элементам базового определяется**

а) ключами доступа: private, public, protected в теле производного класса

б) только ключом доступа protected в заголовке объявления производного класса

с) ключами доступа: private, public, protected в заголовке объявления производного класса

д) ключами доступа: private, public, protected в теле базового класса

**16. Шаблон функции - это...**

а) определение функции, в которой типу обрабатываемых данных присвоено условное обозначение

б) прототип функции, в котором вместо имен параметров указан условный тип

с) определение функции, в котором указаны возможные варианты типов обрабатываемых параметров

д) определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

**17. Выберите правильные утверждения:**

а) по умолчанию члены класса имеют атрибут private

б) по умолчанию члены класса имеют атрибут public;

с) члены класса имеют доступ только к элементам public;

д) элементы класса с атрибутом private доступны только членам класса

**18. Переопределение операций имеет вид:**

а) имя\_класса, ключевое слово operation, символ операции

б) имя\_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы

с) имя\_класса, ключевое слово operator, список аргументов

д) имя\_класса, два двоеточия, ключевое слово operator, символ операции

**19. Полиморфизм – это :**

а) средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов;

б) средство, позволяющее в одном классе использовать методы с одинаковыми именами;

с) средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий

д) средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

**20. Полиморфизм реализован через механизмы:**

а) перегрузки функций, виртуальных функций, шаблонов

б) перегрузки функций, наследования методов, шаблонов;

с) наследования методов, виртуальных функций, шаблонов

д) перегрузки функций, наследования, виртуальных функций.

**21. Без какой библиотеки не будет работать ввод/вывод данных?**

а) math;

б) random;

с) iostream;

д) string;

**22. какая команда используется для ввода/вывода данных в C++?**

а) readln /Writeln ;

б) cin/cout;

с) alert/ prompt

## **Вопросы для подготовки к экзамену/зачёту**

1. Валидация(аттестация) и верификация
2. Общие вопросы верификации ПО. Цели и задачи верификации ПО
3. Статические и динамические методы верификации ПО
4. Факторы и атрибуты внешнего и внутреннего качества ПО
5. Виды и методы тестирования. Смоук-тестирование, регрессионное тестирование, тестирование белого и черного ящиков
6. Тестирование нефункциональных требований
7. Взаимосвязь разработки и тестирования. V-модель разработки ПО.
8. Уровни тестирования. Модульное (unit), интеграционное (integration),
9. Системное (system), приемочное (acceptance) тестирование.
10. Техники тест дизайна. Разбиение на классы эквивалентности и тестирование граничных значений.
11. Понятие дефекта. Основные определения и классификация дефектов.
12. Описание дефектов
13. Атрибуты дефектов. Приоритет(priority) и серьезность(severity) дефектов 14. Определение серьезности дефекта по его описанию — практическое задание 15. Версионирование ПО на разных стадиях разработки.
16. Инструментальные средства поддержки тестирования. Системы отслеживания ошибок (Bug Tracking Systems)
- 17.Arteфакты разработки ПО, относящиеся к тестированию. Тест-кейсы (test cases)
18. Arteфакты разработки ПО, относящиеся к тестированию. План тестирования (test plan).

## **4.3. Типовые задания для оценки освоения МДК.01.03 Разработка мобильных приложений**

### **Вопросы для подготовки к экзамену/зачёту 1.**

Понятие защищенной информационной системы.

2. Свойства защищенной ОС.
3. Безопасность информационных систем в нормативных документах.
4. Классификация защищенности ОС по международным стандартам.
5. Политика безопасности, формальное представление политик.
6. Классификация изъянов защиты.
7. Категории изъянов защиты в ОС.
8. Мобильное программирование, платформы для разработки.
9. Основы работы с сенсорным вводом. Обработка нескольких касаний. Использование изображений.
10. Акселерометр и служба определения местоположения, вторичные потоки выполнения, обработка асинхронных операций и доступ к Веб-сервисам.
11. Особенности использования pivot и panorama.
12. Краткая история ОС Android.

13. Intel для Android: партнерство и инструментарий разработчика.
14. Архитектура приложений для Android. Ресурсы приложения. Пользовательский интерфейс. Инструментарий разработки приложений для Android.
15. Обзор шагов разработки типичного приложения под Android. Особенности разработки с использованием эмулятора. Отладка кода в эмуляторе и на реальных приложениях.  
Пример простейших программ Android-приложения. Запуск приложения на эмуляторе.
16. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS).
17. Планирование покадровой анимации, анимирование, анимация шаблонов, видов, использование класса Camera.
18. Проверка безопасности, работа со службами, основанными на местоположении, использование HTTP-служб, службы AIDL.
19. Подготовка AndroidManifest.xml для загрузки, локализация приложения, подготовка ярлыка приложения, подготовка APK-файла для загрузки, работа пользователя с Android Market.
20. Работа с инструментами Intel для оптимизации отладки Android-приложений

### **Примеры тестовых заданий для текущего контроля МДК 01.03. Разработка мобильных приложений**

1. Ядро какой операционной системы использовалось в качестве базы для ОС Android?
  - (a) Linux
  - (b) OS/2
  - (c) Mac OS
  - (d) Windows
  
2. Большинство приложений для Android написано на языке программирования
  - (a) C++
  - (b) C#
  - (c) Java
  - (d) JavaScript
  
3. Какая графическая библиотека входит в набор библиотек ОС Android?
  - (a) OpenCV
  - (b) DirectX
  - (c) OpenCL
  - (d) Open GL

4. Приложения, не имеющие GUI и выполняющиеся в фоновом режиме – это

- (a) Services
- (b) Intents
- (c) Content Providers
- (d) Activities

5. Какая папка в структуре Android-приложения содержит файлы с исходным кодом на языке Java?

- (a) gen
- (b) src
- (c) bin
- (d) res

6. Приложение какого вида имеет смысл использовать для отображения динамической информации, такой как заряд батареи, прогноз погоды, дата и время?

- (a) фоновое приложение
- (b) приложение переднего плана
- (c) виджет
- (d) смешанное приложение

7. Что находится в папке gen?

- (a) ресурсы приложения
- (b) аудио файлы
- (c) изображения
- (d) автоматически сгенерированные java-файлы

8. При загрузке приложения первой появляется активность,

- (a) чье описание является первым в лексикографическом порядке
- (b) чье описание находится первым в манифесте
- (c) чье описание находится последним в манифесте

(d) правильный вариант ответа отсутствует

9. При проектировании окон приложения необходимо:

- (a) интегрировать в рабочее пространство инструменты, используемые наиболее часто
- (b) для каждой функции делать отдельное диалоговое окно
- (c) все вышеперечисленное
- (d) позволять диалоговому окну оперировать только с одной функцией

10. Какой вид компоновки определяет табличный способ расположения компонентов графического интерфейса пользователя в приложениях под Android?

- (a) RowLayout
- (b) TableLayout
- (c) GridBagLayout
- (d) GridLayout

11. Какой метод вызывается системой в случае, когда активность теряет фокус?

- (a) onPause()
- (b) onRestart()
- (c) onDestroy()
- (d) onStop()

12. Какой движок баз данных используется в ОС Android?

- (a) InnoDB
- (b) DBM
- (c) MyISAM
- (d) SQLite

13. Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий?
- (a) GUI
  - (b) UIComponent
  - (c) View
  - (d) Widget
14. Каждый приемник широковещательных сообщений является наследником класса ...
- (a) ContentProvider
  - (b) BroadcastReceiver
  - (c) ViewReceiver
  - (d) IntentReceiver
15. Какой компонент управляет распределенным множеством данных приложения?
- (a) сервис (Service)
  - (b) активность (Activity)
  - (c) приемник широковещательных сообщений (Broadcast Receiver)
  - (d) контент-провайдер (Content Provider)
16. Какой метод запускает новую активность?
- (a) startActivity()
  - (b) beginActivity()
  - (c) intentActivity()
  - (d) newActivity()
17. Что необходимо сделать при добавлении в проект новой активности?
- (a) скачать и установить специальный инструмент MultiActivity SDK
  - (b) прописать в манифесте информацию о новой активности
  - (c) создать новый проект
  - (d) запустить эмулятор

18. При настройке обратной совместимости необходимо добавить в файл манифеста следующую информацию:

- (a) только минимальную версии Android SDK
- (b) только основную (целевую) версии Android SDK
- (c) минимальную и основную (целевую) версии Android SDK
- (d) информацию о подключенной библиотеке

19. С помощью какого класса можно создавать всплывающие сообщения ?

- (a) PopUp
- (b) Messages
- (c) Help
- (d) Toast

20. Какова главная проблема при проектировании интерфейсов для мобильных устройств?

- (a) небольшой размер экрана
- (b) плохая цветопередача экрана
- (c) энергопотребление
- (d) низкая контрастность экрана

### *Ответы*

1. a) 2. c) 3. d) 4. a) 5. b) 6. c) 7. d) 8. b) 9. a) 10. b) 11. a) 12. d) 13. c)  
14. b) 15. d) 16. a) 17. b) 18. c) 19. d) 20. a)

### **Примеры практических заданий**

МДК 01-03. Разработка мобильных приложений

Веб-приложения являются простым частным случаем мобильных приложений, которые запускаются в браузере. Были рассмотрены следующие примеры программ на языке HTML с фрагментами на JavaScript.

1. Вводятся 3 целых числа и вычисляется минимальное из них и средняя величина.

```
<html>
<head>
<title>Нахождение среднего</title>
<script>
function funk1() {

let a=parseInt(document.getElementById("A").value);
let b=parseInt(document.getElementById("B").value);
let c=parseInt(document.getElementById("C").value);

let res = (a+b+c)/3;
//alert(Math.min(a,b))
document.getElementById("answer").innerHTML = "Среднее: " + res;
document.getElementById("answer").innerHTML += "<br>Наименьшее: " + Math.min(a,b,c);

}
</script>
</head>
<body>

<input type="number" id="A">
<br>
<input type="number" id="B">
<br>
<input type="number" id="C">
<br><br>
<input type="button" value="Решить задачу" onclick="funk1();">
<br><br>

<p id="answer"></p>

</body>
</html>
```

2. Вводятся целые числа в одну строку через пробел и вычисляется минимальное из них и средняя величина.

```
<html>
<head>
```

```

<title>Нахождение среднего</title>
<script>
function funk1() {

let txt = document.getElementById("A").value;

let newtext = txt.replace(/ +/g, ' ');
newtext = newtext.trim();
let words = newtext.split(" ");
//alert(newtext.length);
let N = words.length, S = 0;

for (i=0; i<N; i++)
{
b = parseInt(words[i]);
S += b;
}

let minVal = parseInt(words[0]);
for (i=1; i<N; i++)
{ minVal = Math.min(minVal, parseInt(words[i])); }

let res = S/N;
//alert(Math.min(a,b))
document.getElementById("answer").innerHTML = "Среднее: " + res;
document.getElementById("answer").innerHTML += "<br>Наименьшее: " + minVal;

}
</script>

</head>
<body>

<input type="text" id="A">
<br><br>
<input type="button" value="Решить задачу" onclick="funk1();" />
<br><br>

<p id="answer"></p>

</body>
</html>

```

Введение в язык Java.

Для проведения практических занятий и написания приложений на Android, так же как и для работы с уже написанными кодами необходимо знание основ языка Java.

Следующие примеры программ на языке Java помогают понять различные элементы синтаксиса языка и создают лишь настольные приложения (на ПК), которые напрямую нельзя превратить в мобильные.

Рассматривались следующие примеры с запуском, тестированием, синтаксическим анализом программ. Можно использовать различные IDE (как Visual Studio), так и компилятор `javac` в режиме командной строки для запуска программ.

### 1. Работа со строками, циклом `for` и выводом в Java.

```
public class Welcome
{
    public static void main (String[ ] args)
    {
        String[] greeting = new String[3];
        greeting[0] = "Welcome to Core Java";
        greeting[1] = "by Cay Hortsman";
        greeting[2] = "and Gary Cornell";

        for (String g : greeting)
            System.out.println(g);
    }
}
```

### 2. Работа со строками и заменой в них, с вводом-выводом в Java.

```
import java.io.*;
import java.util.*;
public class WordExchange{
    public static void main(String[] args) throws IOException {
        BufferedReader br=new BufferedReader(new
        InputStreamReader(System.in, "CP866"));
        String s, w1, w2;
        int i=0;
        System.out.print("Enter the sentence: ");
        s=br.readLine();
```

```

System.out.print("Enter word for search:");
w1=br.readLine();
System.out.print("Enter word for replacement:");
w2=br.readLine();
//первый вариант
i=s.indexOf(w1); //Находим номер символа в строке s
//с которого подстрока w1 //входит
//в строку s. (-1 если такой нет).
if(i != -1){
s=s.substring(0,i)+w2+s.substring(i+w1.length());
}
//второй вариант
s=s.replaceFirst(w1,w2);
//
System.out.println("Resulting sentence is: "+s);
}
}

```

### 3. Работа с классами, изображениями и окнами в Java.

```

import java.awt.EventQueue;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;

/**
 * Программа для просмотра изображений
 */
public class ImageViewer
{
public static void main(String[ ] args)
{
EventQueue.invokeLater (new Runnable ( )
{
public void run()
{
JFrame frame = new ImageViewerFrame ( ) ;
frame.setTitle ("ImageViewer");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible (true);
}
}
}
}

```

```

    }
    } );
}
}

/**
 * Фрейм с текстовой меткой для вывода изображения.
 */
class ImageViewerFrame extends JFrame
{
private JLabel label;
private JFileChooser chooser;
private static final int DEFAULT_WIDTH = 300;
private static final int DEFAULT_HEIGHT = 400;
public ImageViewerFrame ()
{
setSize (DEFAULT_WIDTH, DEFAULT_HEIGHT) ;

// использовать метку для вывода изображений на экран
label = new JLabel();
add(label);

// установить селектор файлов
chooser = new JFileChooser () ;
chooser.setCurrentDirectory (new File(".") ) ;

// установить строку меню
JMenuBar menuBar = new JMenuBar();
setJMenuBar(menuBar) ;

JMenu menu = new JMenu ("File");
menuBar.add(menu);

JMenuItem openItem = new JMenuItem("Open");
menu.add(openItem);
openItem.addActionListener (new ActionListener ()
{
public void actionPerformed(ActionEvent event)
{
// отобразить диалоговое окно селектора файлов
int result = chooser.showOpenDialog(null);

```

```
// если файл выбран, задать его в качестве пиктограммы для метки
if (result == JFileChooser.APPROVE_OPTION)
{
String name = chooser.getSelectedFile () .getPath();
label.setIcon(new ImageIcon(name) );
}
}
});
```

```
JMenuItem exitItem = new JMenuItem("Exit");
menu.add(exitItem);
exitItem.addActionListener (new ActionListener ()
{
public void actionPerformed(ActionEvent event)
{
System.exit (0);
}
});

}
}
```

#### 4. Ввод-вывод и арифметика в Java.

```
import java.util.Scanner;// импортируем класс Scanner

public class Main {// объявляется класс Main

    public static void main(String[] args) {// так включаются программы,

        Scanner sc = new Scanner(System.in);
        System.out.println("Введите число:");
        int a = sc.nextInt();
        sc.close();
        System.out.println("Квадрат числа:" + (a * a));

    }
}
```

#### 5. Ввод-вывод и арифметика в Java.

```
import java.util.Scanner;// импортируем класс Scanner
```

```
public class Main { // объявляется класс Main
```

```
    public static void main(String[] args) { // так включаются программы,
```

```
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите число:");
        int a = -sc.nextInt();
        sc.close();
        System.out.println("Обратное число:" + a);
```

```
    }
}
```

6. Ввод-вывод и арифметика в Java.

```
import java.util.Scanner; // импортируем класс Scanner
```

```
public class Main { // объявляется класс Main
```

```
    public static void main(String[] args) { // так включаются программы,
```

```
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите число:");
        int a = sc.nextInt();
        System.out.println("Введите число:");
        int b = sc.nextInt();
        sc.close();
        System.out.println("Разность:" + (a - b));
```

```
    }
}
```

7. Ввод-вывод и случайные числа в Java.

```
public class Test { // объявляется класс Test
```

```
    public static void main(String[] args) {
```

```
        int i = (int) (Math.random() * 10 + 1);
```

```
        System.out.println("Случайное число:" + i);  
    }  
}
```

#### 8. Ввод-вывод и циклы в Java.

```
import java.util.Scanner;  
  
public class Test {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Введите число:")  
        int a = sc.nextInt();  
        sc.close();  
  
        int i = 0;  
        do {  
            System.out.println("Повтор:" + i);  
            i = i + 1;  
        } while (i < (a + 1));  
    }  
}
```

#### 9. Ввод-вывод, циклы и случайные числа в Java.

```
import java.util.Scanner;  
  
public class Game {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        int a;  
        int result = (int) (Math.random() * 10 + 1);  
  
        do {  
            System.out.println("Угадайте число:");  
            a = sc.nextInt();  
        } while (a != result);  
    }  
}
```

```

} while (result != a);

System.out.println("Ответ:" + a);
sc.close();
}
}

```

#### 10. Ввод-вывод и условный оператор в Java.

```
import java.util.Scanner;// импортируем класс Scanner
```

```

public class Task {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Введите число:");
        int a = sc.nextInt();
        System.out.println("Введите число:");
        int b = sc.nextInt();
        sc.close();

        if (a != b) {
            if (a > b) {
                System.out.println(a + " больше " + b);
            } else {
                System.out.println(b + " больше " + a);
            }
        } else {
            System.out.println("Числа равны");
        }
    }
}

```

#### 11. Ввод-вывод и математические операторы в Java.

```

public class Test {
    public static void main(String[] args) {
        // библиотека MATH, тригонометрия
        double a = 0.5;
        double b, c, d, e, f, g;
    }
}

```

```

b = Math.sin(a);
c = Math.cos(a);
d = Math.tan(a);
e = Math.asin(a);
f = Math.acos(a);
g = Math.atan(a);
System.out.println("sin " + a + " = " + b);
System.out.println("cos " + a + " = " + c);
System.out.println("tan " + a + " = " + d);
System.out.println("asin " + a + " = " + e);
System.out.println("acos " + a + " = " + f);
System.out.println("atan " + a + " = " + g);
}
}

```

## 12. Ввод-вывод и математические операторы в Java.

```
import java.util.Scanner;
```

```

public class Main {
    public static void main(String[] args) {
        // библиотека MATH, hypot
        Scanner sc = new Scanner(System.in);
        double a;
        double b;
        System.out.println("Введите число:");
        a = sc.nextInt();
        System.out.println("Введите число:");
        b = sc.nextInt();
        sc.close();
        double g = Math.hypot(a, b);
        System.out.print("Гипотенуза треугольника с катетами ");
        System.out.println(a + " и " + b + " = " + g);
    }
}

```

## 13. Ввод-вывод и строковые функции в Java.

```
import java.util.Scanner;
```

```

public class Task {
    public static void main(String[] args) {
        // методы класса String, length()

        Scanner sc = new Scanner(System.in);
        String adres;
        System.out.println("Введите слово:");
        adres = sc.nextLine();
        sc.close();

        int numer = adres.length();
        System.out.println("Длина строки:" + numer);
    }
}

```

14. *Ввод-вывод и строковые функции в Java.*

```

import java.util.Scanner;

```

```

public class Task {
    public static void main(String[] args) {
        // методы класса String, equals
        Scanner sc = new Scanner(System.in);
        String target = "аккумулятор";
        System.out.println("В слове 'акумулятор' есть ошибки, исправьте ошибки.");
        System.out.println("Введите слово:");
        String word = sc.nextLine();
        sc.close();

        if (word.equals(target)) {
            System.out.println(word + " правильно");
        } else {
            System.out.println(word + " не правильно");
        }
    }
}

```

15. *Ввод-вывод, строковые функции, циклы и условный оператор в Java.*

```

import java.util.Scanner;

```

```

public class Main {
    public static void main(String[] args) {
        // методы класса String, сортировка по алфавиту
        // compareTo
        Scanner sc = new Scanner(System.in);
        String a[] = new String[3];
        System.out.println("Введите слово:");
        a[0] = sc.nextLine();
        System.out.println("Введите слово:");
        a[1] = sc.nextLine();
        System.out.println("Введите слово:");
        a[2] = sc.nextLine();
        sc.close();
        for (int j = 0; j < a.length; j++) {
            for (int i = j + 1; i < a.length; i++) {
                if (a[i].compareTo(a[j]) < 0) {
                    String helper = a[j];
                    a[j] = a[i];
                    a[i] = helper;
                }
            }
            System.out.println(a[j]);
        }
    }
}

```

16. *Ввод-вывод, строковые функции и условный оператор в Java.*

```

import java.util.Scanner;

```

```

public class Main {
    public static void main(String[] args) {
        // методы класса String, isEmpty
        Scanner sc = new Scanner(System.in);
        String a;
        System.out.println("Введите слово:");
        a = sc.nextLine();
        sc.close();
        if (a.isEmpty()) {
            System.out.println("Пусто");
        }
    }
}

```

```
    } else {  
        System.out.println(a);  
    }  
}  
}
```

### 17. Работа с классами в Java.

```
public class Box {  
    int width;  
    int height;  
    int length;  
  
    public void valueOfBox() {  
        System.out.println("Объем = " + width * height * length);  
    }  
  
    public void lenghtOfSides() {  
        System.out.println("Длина сторон = " + (width+height+length)*4);  
    }  
}
```

---

```
import java.util.Scanner;
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Box box1 = new Box();  
  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Введите стороны коробки");  
        System.out.println("Введите число:");  
        box1.width = sc.nextInt();  
        System.out.println("Введите число:");  
        box1.height = sc.nextInt();  
        System.out.println("Введите число:");  
        box1.lenght = sc.nextInt();  
    }  
}
```

```

box1.valueOfBox();
box1.lenghtOfSides();

sc.close();
}

```

## 18. Оконные приложения в Java.

```

import java.awt.BorderLayout;
import javax.swing.*;    // подключаем все средства java Swing
public class Program { // класс с методом main()
    public static void main(String[] args) {
        JFrame frame = new JFrame("My First GUI");
        // задаём путь к иконке (т.е. где находится файл)
        String path = "../images/icon.png";
        // создание через обращение к главному классу приложения и его ресурсам
        ImageIcon icon = new ImageIcon(Program.class.getResource(path));
        // помещение иконки на frame
        frame.setIconImage(icon.getImage());
        JPanel buttonsPanel = new JPanel();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // задание геометрии окна
        frame.setSize(300, 300);
        // запрет изменения размеров
        frame.setResizable(false);
        frame.setLocationRelativeTo(null);
        JButton start = new JButton("Старт");
        JButton stop = new JButton("Стоп");
        buttonsPanel.add(start);
        buttonsPanel.add(stop);
        frame.getContentPane().add(BorderLayout.NORTH, buttonsPanel);
        frame.setVisible(true);
    }
}

```

В следующих примерах исходных кодов для мобильных приложений на Android нужно провести синтаксический разбор программ на языке Java и XML (без подробного анализа самих алгоритмов).

Данные задания задавались также и на экзамене.

1.

### Main Activity

```
package com.example.menugroup;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
}
```

### main

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
```

```
    <item
```

```
        android:id="@+id/menu_bookmark"
```

```
        android:icon="@drawable/icon_bookmark"
```

```
        android:title="Save"/>
```

```
    <!-- menu group -->
```

```
    <group android:id="@+id/group_delete" >
```

```
        <item
```

```
            android:id="@+id/menu_more"
```

```
            android:title="More"/>
```

```
        <item
```

```
            android:id="@+id/menu_delete"
```

```
            android:title="Delete"/>
```

```
    </group>
```

</menu>

2.

### Main Activity

```
package com.example.usingcheckablemenuitems;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.view.Menu;
```

```
import android.view.MenuItem;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
    /*
```

```
     * When a checkable item is selected, the system calls your respective
```

```
     * item-selected callback method (such as onOptionsItemSelected()). It is
```

```
     * here that you must set the state of the checkbox, because a checkbox or
```

```
     * radio button does not change its state automatically. You can query the
```

```
     * current state of the item (as it was before the user selected it) with
```

```
     * isChecked() and then set the checked state with setChecked()
```

```
     */
```

```
    @Override
```

```
    public boolean onOptionsItemSelected(MenuItem item) {
```

```
        switch (item.getItemId()) {
```

```
            case R.id.save:
```

```
            case R.id.delete:
```

```
                if (item.isChecked())
```

```
                    item.setChecked(false);
```

```

        else
            item.setChecked(true);
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}

```

## Menu

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <group android:checkableBehavior="single" >
        <item
            android:id="@+id/delete"
            android:title="Delete"/>
        <item
            android:id="@+id/save"
            android:title="Save"/>
    </group>

</menu>

```

3.

## Main Activity

```

package com.example.datepickerdialog;

import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.DatePickerDialog.OnDateSetListener;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_main);
Button but = (Button) findViewById(R.id.button1);
but.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        DatePickerDialog datePickerDialog = new DatePickerDialog(
            MainActivity.this, new OnDateSetListener() {
                public void onDateSet(DatePicker view, int year,
                    int monthOfYear, int dayOfMonth) {
                    // TODO Use the selected date.
                }
            }, 2013, 10, 11);
        datePickerDialog.show();
    }
});
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

```

4.

### MainActivity.java

```

package com.example.progressdialog;

import android.app.Activity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    Button button = (Button) findViewById(R.id.button1);  
    button.setOnClickListener(new View.OnClickListener() {  
        ProgressDialog progressdialog;
```

@Override

```
public void onClick(View v) {  
    // TODO Auto-generated method stub  
    progressdialog = new ProgressDialog(MainActivity.this);  
    progressdialog.setMessage("Loading....");  
    progressdialog.setTitle("Please Wait..");  
    progressdialog.setProgressStyle(progressdialog.STYLE_HORIZONTAL);  
    progressdialog.setProgress(0);  
    progressdialog.setMax(100);  
    progressdialog.show();  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            // TODO Auto-generated method stub  
            try {  
                while (progressdialog.getProgress() <= progressdialog.getMax()) {  
                    Thread.sleep(1000);  
                    handle.sendMessage(handle.obtainMessage());  
                    if (progressdialog.getProgress() == progressdialog.getMax()) {  
                        progressdialog.dismiss();  
                    }  
                }  
            } catch (Exception e) {  
            }  
        }  
    }).start();  
}
```

```
Handler handle = new Handler() {  
    @Override  
    public void handleMessage(Message msg) {  
        // TODO Auto-generated method stub  
        super.handleMessage(msg);  
        progressdialog.incrementProgressBy(20);
```

```

    }
};
});
}

```

**@Override**

```

public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

```

```

}

```

### **Activity\_Main.xml**

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

```

**<Button**

```

    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignRight="@+id/textView1"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="80dp"
    android:text="Launch" />

```

```

</RelativeLayout>

```

5.

```

package com.example.create_menu1;

```

```

import android.os.Bundle;

```

```

import android.app.Activity;

```

```
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        super.onCreateOptionsMenu(menu);
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
}
```

**Xml**

**Main**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    tools:context=".MainActivity" >
```

```
    <TextView
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="40dp"
```

```
        android:text="A Menu Activity"
```

```
        android:textSize="30dp"
```

```
        android:layout_marginTop="120dp"
```

```
        android:gravity="center_vertical|center_horizontal" />
```

```
</RelativeLayout>
```

**Menu**

```

<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/Menu_label1"
        android:alphabeticShortcut="1"
        android:title="Menu_label"/>

    <item
        android:id="@+id/Menu_label2"
        android:alphabeticShortcut="2"
        android:title="Menu_label2"/>

</menu>

```

6.

### Main Activity

```

package com.example.creating_listview;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button)findViewById(R.id.button1);
        button.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                startActivity(new Intent("com.example.creating_listview.list"));
            }
        });
    }
}

```

```
    }  
  });  
}
```

**@Override**

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}
```

**@Override**

```
protected void onPause() {  
    // TODO Auto-generated method stub  
    super.onPause();  
    finish();  
}
```

```
}
```

## List

```
package com.example.creating_listview;
```

```
import android.app.ListActivity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.AdapterView;
```

```
import android.widget.ListView;
```

```
public class list extends ListActivity {
```

```
    String list_item[] = { "MainActivity", "FirstActivity", "SecondActivity" };
```

**@Override**

```
protected void onCreate(Bundle savedInstanceState) {  
    // TODO Auto-generated method stub  
    super.onCreate(savedInstanceState);  
    setListAdapter(new ArrayAdapter<String>(this,  
        android.R.layout.simple_expandable_list_item_1, list_item));  
}
```

@Override

```
protected void onItemClick(ListView l, View v, int position, long id) {
```

```
    // TODO Auto-generated method stub
```

```
    super.onItemClick(l, v, position, id);
```

```
    String open_class = list_item[position];
```

```
    try {
```

```
        Class selected = Class.forName("com.example.creating_listview."+open_class);
```

```
        Intent selected_activity = new Intent(this,selected);
```

```
        startActivity(selected_activity);
```

```
    } catch (ClassNotFoundException e) {
```

```
        // TODO Auto-generated catch block
```

```
        e.printStackTrace();
```

```
    }
```

```
    finally {
```

```
    }
```

```
}
```

```
}
```

**FirstActivity**

```
package com.example.creating_listview;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
public class FirstActivity extends Activity{
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        // TODO Auto-generated method stub
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.first);
```

```
    }
```

```
}
```

**SecondActivity**

```
package com.example.creating_listview;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
public class SecondActivity extends Activity{
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        // TODO Auto-generated method stub
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.second);
```

```
    }
```

```
}
```

```
Xml
```

```
Main
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    tools:context=".MainActivity" >
```

```
<TextView
```

```
    android:id="@+id/textView1"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="40dp"
```

```
    android:text="A List View Activity"
```

```
    android:layout_marginTop="100dp"
```

```
    android:gravity="center_vertical|center_horizontal"
```

```
    android:textSize="30dp" />
```

```
<Button
```

```
    android:id="@+id/button1"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignLeft="@+id/textView1"
```

```
    android:layout_alignParentBottom="true"
```

```
    android:layout_alignParentRight="true"
```

```
android:layout_marginBottom="88dp"
android:text="Launch List View Activity" />
```

**</RelativeLayout>**

**First**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@android:color/black">
```

**<TextView**

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="60dp"
    android:gravity="center"
    android:text="First Activity"
    android:layout_marginRight="40dp"
    android:textColor="@android:color/holo_blue_dark"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textSize="60dp" />
```

**</LinearLayout>**

**Second**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@android:color/black">
```

**<TextView**

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="60dp"
    android:gravity="center"
```

```
    android:text="Second Activity"
    android:layout_marginRight="40dp"
    android:textColor="@android:color/holo_blue_dark"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textSize="60dp" />
```

**</LinearLayout>**

**AndroidManifest**

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.creating_listview"
    android:versionCode="1"
    android:versionName="1.0" >
```

**<uses-sdk**

```
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />
```

**<application**

```
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
```

**<activity**

```
    android:name="com.example.creating_listview.MainActivity"
    android:label="@string/app_name" >
```

**<intent-filter>**

```
    <action android:name="android.intent.action.MAIN" />
```

```
    <category android:name="android.intent.category.LAUNCHER" />
```

**</intent-filter>**

**</activity>**

**<activity**

```
    android:name="com.example.creating_listview.list"
    android:label="@string/app_name" >
```

**<intent-filter>**

```
    <action android:name="com.example.creating_listview.list" />
```

```
    <category android:name="android.intent.category.DEFAULT" />
```

**</intent-filter>**

**</activity>**

```

<activity
    android:name="com.example.creating_listview.FirstActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="com.example.creating_listview.FirstActivity" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity
    android:name="com.example.creating_listview.SecondActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="com.example.creating_listview.SecondActivity" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

</application>

</manifest>

```

7.

### Main Activity

```
package com.example.options_menu;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```

public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(1, 1, 0, "Item1");
    menu.add(1, 2, 1, "Item2");
    menu.add(1, 3, 2, "Item3");
    menu.add(1, 4, 3, "Item4");
    menu.add(1, 5, 4, "Item5");
    return true;
}
}

```

8.

### Main Activity

```

package com.example.createmenu;

```

```

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

```

```

public class MainActivity extends Activity {

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

```

```

    @Override

```

```

    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.layout.menu, menu);
        return true;
    }

```

```

    @Override

```

```

    public boolean onOptionsItemSelected(MenuItem item) {

        switch (item.getItemId()) {

            case R.id.save:

```

```
Toast.makeText(MainActivity.this, "Save Menu Selected",  
    Toast.LENGTH_SHORT).show();  
return true;
```

```
case R.id.search:
```

```
    Toast.makeText(MainActivity.this, "Search Menu Selected",  
        Toast.LENGTH_SHORT).show();  
return true;
```

```
case R.id.share:
```

```
    Toast.makeText(MainActivity.this, "Share Menu Selected",  
        Toast.LENGTH_SHORT).show();  
return true;
```

```
case R.id.delete:
```

```
    Toast.makeText(MainActivity.this, "Delete Menu Selected",  
        Toast.LENGTH_SHORT).show();  
return true;
```

```
default:
```

```
    return super.onOptionsItemSelected(item);
```

```
    }
```

```
    }
```

```
    }
```

```
menu
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
```

```
<item
```

```
    android:id="@+id/save"
```

```
    android:icon="@drawable/icon_save"
```

```
    android:title="Save"/>
```

```
<item
```

```
    android:id="@+id/search"
```

```
    android:icon="@drawable/icon_search"
```

```
    android:title="Search"/>
```

```
<item
```

```
    android:id="@+id/share"
```

```
    android:icon="@drawable/icon_share"
```

```
android:title="Share"/>
```

```
<item
```

```
    android:id="@+id/delete"
```

```
    android:icon="@drawable/icon_delete"
```

```
    android:title="Delete"/>
```

```
</menu>
```

9.

### Main Activity

```
package com.example.text_view_rounded;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.Toast;
```

```
import android.widget.VideoView;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Button but = (Button) findViewById(R.id.button1);
```

```
        but.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                // TODO Auto-generated method stub
```

```
                Toast.makeText(getApplicationContext(), "On Click",
```

```
                    Toast.LENGTH_LONG).show();
```

```
            }
```

```
        });
```

```
    }
```

```
    @Override
```

```

public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

```

```

}

```

### mystyle

```

<?xml version="1.0" encoding="utf-8"?>

```

```

<selector xmlns:android="http://schemas.android.com/apk/res/android">

```

```

    <item android:state_focused="true" android:state_pressed="true"><shape>

```

```

        <stroke android:width="2.3dp" android:color="#a7bf42" />

```

```

        <corners android:radius="10dp" />

```

```

    </shape></item>

```

```

    <item android:state_focused="false" android:state_pressed="true"><shape>

```

```

        <stroke android:width="2.3dp" android:color="#a7bf42" />

```

```

        <corners android:radius="10dp" />

```

```

    </shape></item>

```

```

    <item android:state_focused="true" android:state_pressed="false"><shape>

```

```

        <stroke android:width="0.7dp" android:color="#000000" />

```

```

        <corners android:radius="15dp" />

```

```

    </shape></item>

```

```

    <item android:state_focused="false" android:state_pressed="false"><shape>

```

```

        <gradient android:angle="270" android:centerColor="#F6EBC9" android:endColor="#a7bf42"
android:startColor="#a7bf42" />

```

```

        <stroke android:width="0.7dp" android:color="#000000" />

```

```

        <corners android:radius="15dp" />

```

```

    </shape></item>

```

```

    <item android:state_enabled="true"><shape>

```

```

        <padding android:bottom="4dp" android:left="4dp" android:right="4dp" android:top="4dp" />

```

```

    </shape></item>

```

```

</selector>

```

### Activity\_Main

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

```
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:paddingBottom="@dimen/activity_vertical_margin"  
android:paddingLeft="@dimen/activity_horizontal_margin"  
android:paddingRight="@dimen/activity_horizontal_margin"  
android:paddingTop="@dimen/activity_vertical_margin"  
tools:context=".MainActivity" >
```

### <Button

```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentLeft="true"  
android:layout_alignParentRight="true"  
android:layout_centerVertical="true"  
android:text="Round Cornered Button"  
android:background="@drawable/mystyle"/>
```

### </RelativeLayout>

10.

### Main Activity

```
package com.example.read_sms;
```

```
//inspired from www.stack overflow.com and www.vogella.com/android/
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import android.app.ListActivity;
```

```
import android.content.ContentResolver;
```

```
import android.database.Cursor;
```

```
import android.net.Uri;
```

```
import android.os.Bundle;
```

```
import android.provider.BaseColumns;
```

```
import android.provider.ContactsContract;
```

```
import android.util.Log;
```

```
import android.view.View;
```

```
import android.widget.ListView;
```

```
import android.widget.Toast;
```

```

public class MainActivity extends ListActivity {

    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        List<SmsFormat> List = new ArrayList<SmsFormat>();

        Uri uri = Uri.parse("content://sms/inbox");
        Cursor cur = getContentResolver().query(uri, null, null, null, null);
        startManagingCursor(cur);

        if (cur.moveToFirst()) {
            for (int i = 0; i < cur.getCount(); i++) {
                SmsFormat sms = new SmsFormat();
                if (cur.getColumnIndexOrThrow("body") > -1) {
                    sms.setBody(cur
                        .getString(cur.getColumnIndexOrThrow("body"))
                        .toString());
                }
                if (cur.getColumnIndexOrThrow("address") > -1) {
                    sms.setNumber(cur.getString(
                        cur.getColumnIndexOrThrow("address")).toString());
                }

                String name = null;
                name = Search(cur.getString(
                    cur.getColumnIndexOrThrow("address")).toString());

                if (name != " ") {
                    sms.setName(name);
                    List.add(sms);
                }
                cur.moveToNext();
            }
        }
    }
}

```

```

cur.close();

setListAdapter(new ListAdapter(this, List));

}

```

**@Override**

```

protected void onItemClick(ListView l, View v, int position, long id) {
    SmsFormat sms = (SmsFormat) getListAdapter().getItem(position);
    Log.i("MSG", sms.getBody());
}

```

**public** String Search(String number) {

```

    Uri uri = Uri.withAppendedPath(
        ContactsContract.PhoneLookup.CONTENT_FILTER_URI,
        Uri.encode(number));
    String name = " ";

    ContentResolver contentResolver = getContentResolver();
    Cursor contactLookup = contentResolver.query(uri, new String[] {
        BaseColumns._ID, ContactsContract.PhoneLookup.DISPLAY_NAME },
        null, null, null);

```

```

try {
    if (contactLookup != null && contactLookup.getCount() > 0) {
        contactLookup.moveToNext();
        name = contactLookup.getString(contactLookup
            .getColumnIndex(ContactsContract.Data.DISPLAY_NAME));
    }
} finally {
    if (contactLookup != null) {
        contactLookup.close();
    }
}

```

```

return name;
}

```

**List Adapter**

```

package com.example.read_sms;

```

```

import java.util.List;

import android.content.Context;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.drawable.BitmapDrawable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.TextView;

public class ListAdapter extends ArrayAdapter<SmsFormat> {

    private final Context context;
    private final List<SmsFormat> smslist;

    public ListAdapter(Context context, List<SmsFormat> smsList) {
        super(context, R.layout.activity_main, smsList);
        this.context = context;
        this.smslist = smsList;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View rowView = inflater.inflate(R.layout.activity_main, parent, false);

        TextView senderNumber = (TextView) rowView.findViewById(R.id.textView1);

        senderNumber.setText(smslist.get(position).getName() + "\n"
            + smslist.get(position).getNumber() + "\n" + "\n"
            + smslist.get(position).getBody());

        return rowView;
    }
}

```

## Sms Format

```
package com.example.read_sms;
```

```
public class SmsFormat {  
  
    private String number;  
    private String body;  
    private String name;  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getNumber() {  
        return number;  
    }  
  
    public void setNumber(String number) {  
        this.number = number;  
    }  
  
    public String getBody() {  
        return body;  
    }  
  
    public void setBody(String body) {  
        this.body = body;  
    }  
  
}
```

## Activity\_Main

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >
```

<TextView

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:text="Medium Text"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

</RelativeLayout>

11.

### Main Activity

```
package com.example.read_sms;
import java.util.ArrayList;
import java.util.List;

import android.app.ListActivity;
import android.content.ContentResolver;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.BaseColumns;
import android.provider.ContactsContract;
import android.util.Log;
import android.view.View;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends ListActivity {

    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
    }
}
```

```
}
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

```
    List<SmsFormat> List = new ArrayList<SmsFormat>();
```

```
    Uri uri = Uri.parse("content://sms/inbox");
```

```
    Cursor cur = getContentResolver().query(uri, null, null, null, null);  
    startManagingCursor(cur);
```

```
    if (cur.moveToFirst()) {
```

```
        for (int i = 0; i < cur.getCount(); i++) {
```

```
            SmsFormat sms = new SmsFormat();
```

```
            if (cur.getColumnIndexOrThrow("body") > -1) {
```

```
                sms.setBody(cur  
                    .getString(cur.getColumnIndexOrThrow("body"))  
                    .toString());
```

```
            }
```

```
            if (cur.getColumnIndexOrThrow("address") > -1) {
```

```
                sms.setNumber(cur.getString(  
                    cur.getColumnIndexOrThrow("address")).toString());
```

```
            }
```

```
            String name = null;
```

```
            name = Search(cur.getString(  
                cur.getColumnIndexOrThrow("address")).toString());
```

```
            if (name != " ") {
```

```
                sms.setName(name);
```

```
                List.add(sms);
```

```
            }
```

```
            cur.moveToNext();
```

```
        }
```

```
    }
```

```
    cur.close();
```

```
    setListAdapter(new ListAdapter(this, List));
```

```
}
```

@Override

```
protected void onItemClick(ListView l, View v, int position, long id) {  
    SmsFormat sms = (SmsFormat) getListAdapter().getItem(position);  
    Log.i("MSG", sms.getBody());  
  
}
```

```
public String Search(String number) {  
    Uri uri = Uri.withAppendedPath(  
        ContactsContract.PhoneLookup.CONTENT_FILTER_URI,  
        Uri.encode(number));  
    String name = " ";  
  
    ContentResolver contentResolver = getContentResolver();  
    Cursor contactLookup = contentResolver.query(uri, new String[] {  
        BaseColumns._ID, ContactsContract.PhoneLookup.DISPLAY_NAME },  
        null, null, null);  
  
    try {  
        if (contactLookup != null && contactLookup.getCount() > 0) {  
            contactLookup.moveToNext();  
            name = contactLookup.getString(contactLookup  
                .getColumnIndex(ContactsContract.Data.DISPLAY_NAME));  
        }  
    } finally {  
        if (contactLookup != null) {  
            contactLookup.close();  
        }  
    }  
  
    return name;  
}
```

### List Adapter

```
package com.example.read_sms;  
  
import java.util.List;  
  
import android.content.Context;  
import android.content.res.Resources;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;
```

```

import android.graphics.Canvas;
import android.graphics.drawable.BitmapDrawable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.TextView;

public class ListAdapter extends ArrayAdapter<SmsFormat> {

    private final Context context;
    private final List<SmsFormat> smslist;

    public ListAdapter(Context context, List<SmsFormat> smsList) {
        super(context, R.layout.activity_main, smsList);
        this.context = context;
        this.smslist = smsList;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View rowView = inflater.inflate(R.layout.activity_main, parent, false);

        TextView senderNumber = (TextView) rowView.findViewById(R.id.textView1);

        senderNumber.setText(smslist.get(position).getName() + "\n"
            + smslist.get(position).getNumber() + "\n" + "\n"
            + smslist.get(position).getBody());

        return rowView;
    }
}

Sms Format

package com.example.read_sms;

public class SmsFormat {

```

```
private String number;  
private String body;  
private String name;
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
public String getNumber() {  
    return number;  
}
```

```
public void setNumber(String number) {  
    this.number = number;  
}
```

```
public String getBody() {  
    return body;  
}
```

```
public void setBody(String body) {  
    this.body = body;  
}
```

```
}
```

### **Activity\_Main**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    tools:context=".MainActivity" >
```

### **<TextView**

```
    android:id="@+id/textView1"
```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:text="Medium Text"
    android:textAppearance="?android:attr/textAppearanceMedium" />

```

</RelativeLayout>

12.

### Main Activity

```
package com.example.drop_down_navigation;
```

```
import android.app.ActionBar;
```

```
import android.os.Bundle;
```

```
import android.support.v4.app.Fragment;
```

```
import android.support.v4.app.FragmentActivity;
```

```
import android.support.v4.app.NavUtils;
```

```
import android.view.Gravity;
```

```
import android.view.LayoutInflater;
```

```
import android.view.Menu;
```

```
import android.view.MenuItem;
```

```
import android.view.View;
```

```
import android.view.ViewGroup;
```

```
import android.widget.AdapterView;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends FragmentActivity implements
```

```
    ActionBar.OnNavigationListener {
```

```
    private static final String item_selected = "navigation_item";
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        // setting the action bar foe a drop down naviagtion
```

```

final ActionBar actionBar = getActionBar();
actionBar.setDisplayShowTitleEnabled(false);
actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_LIST);

actionBar.setListNavigationCallbacks(
    new ArrayAdapter<String>(actionBar.getThemedContext(),
        android.R.layout.simple_list_item_1,
        android.R.id.text1, new String[] {
            getString(R.string.title_section1),
            getString(R.string.title_section2),
            getString(R.string.title_section3), }), this);
}

@Override
public void onRestoreInstanceState(Bundle savedInstanceState) {
    // restoring the previous drop navigations
    if (savedInstanceState.containsKey(item_selected)) {
        getActionBar().setSelectedItem( savedInstanceState.getInt(item_selected));
    }
}

@Override
public void onSaveInstanceState(Bundle outState) {
    outState.putInt(item_selected, getActionBar().getSelectedItem());
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onNavigationItemSelected(int position, long id) {
    Fragment fragment = new myFragement();
    Bundle args = new Bundle();
    args.putInt(myFragement.string, position + 1);
    fragment.setArguments(args);
    getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment).commit();
    return true;
}

```

```

public static class myFragement extends Fragment {
    public static final String string = "number";
    public myFragement() {}

```

```

    @Override

```

```

    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View fragement_view = inflater.inflate(
            R.layout.layout_fragement, container, false);
        TextView text_ = (TextView) fragement_view.findViewById(R.id.section_label);
        text_.setText(Integer.toString(getArguments().getInt(string)));
        return fragement_view;
    }
}

```

### layout\_fragement

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com  ools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity$DummySectionFragment" >

    <TextView
        android:id="@+id/section_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

```

```

</RelativeLayout>

```

### Activity\_Main

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com  ools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:ignore="MergeRootFrame" />

```

13.

### MainActivity.java

```
package com.example.cancel_alarm_intent;
```

```
import java.util.Calendar;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.app.AlarmManager;
```

```
import android.app.PendingIntent;
```

```
import android.content.Context;
```

```
import android.content.Intent;
```

```
import android.view.Menu;
```

```
import android.view.View;
```

```
import android.view.View.OnClickListener;
```

```
import android.widget.Button;
```

```
import android.widget.Toast;
```

```
public class MainActivity extends Activity implements OnClickListener {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Button start = (Button) findViewById(R.id.button1);
```

```
        Button end = (Button) findViewById(R.id.button2);
```

```
        start.setOnClickListener(this);
```

```
        end.setOnClickListener(this);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
    @Override
```

```
    public void onClick(View v) {
```

```

// TODO Auto-generated method stub
Intent intent = new Intent(this, MyService.class);
startService(new Intent(this, MyService.class));
Calendar cal = Calendar.getInstance();
PendingIntent pintent = PendingIntent.getService(this, 0, intent, 0);

AlarmManager alarm = (AlarmManager) getSystemService(Context.ALARM_SERVICE);

switch (v.getId()) {

case R.id.button1:
    // Start service every 20 seconds
    Toast.makeText(getApplicationContext(), "Alarm Evoked",
        Toast.LENGTH_LONG).show();
    alarm.setRepeating(AlarmManager.RTC_WAKEUP, cal.getTimeInMillis(),
        20 * 1000, pintent);

    break;
case R.id.button2:
    Toast.makeText(getApplicationContext(), "Alarm Canceled",
        Toast.LENGTH_LONG).show();
    alarm.cancel(pintent);
    break;
default:
    break;
}
}
}

```

### **MyService.java**

```

package com.example.cancel_alarm_intent;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.util.Log;
import android.widget.Toast;

public class MyService extends Service {

    @Override
    public IBinder onBind(Intent intent) {

```

```

    // TODO Auto-generated method stub
    return null;
}

@Override
public void onCreate() {
    Toast.makeText(this, " MyService Created ", Toast.LENGTH_LONG).show();
}

@Override
public void onStart(Intent intent, int startId) {
    Toast.makeText(this, " MyService Started", Toast.LENGTH_LONG).show();
}

@Override
public void onDestroy() {
    // TODO Auto-generated method stub
    Toast.makeText(this, "Servics Stopped", Toast.LENGTH_SHORT).show();
    super.onDestroy();
}
}

```

### Activity\_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="71dp"
        android:text="Start Stop Alarm Intent"

```

```
android:textAppearance="?android:attr/textAppearanceLarge" />
```

### **<Button**

```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentLeft="true"  
android:layout_below="@+id/textView1"  
android:layout_marginTop="108dp"  
android:text="Start" />
```

### **<Button**

```
android:id="@+id/button2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/button1"  
android:layout_alignBottom="@+id/button1"  
android:layout_alignRight="@+id/textView1"  
android:text="Cancel" />
```

### **</RelativeLayout>**

#### **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
package="com.example.cancel_alarm_intent"  
android:versionCode="1"  
android:versionName="1.0" >  
  
  <uses-sdk  
    android:minSdkVersion="8"  
    android:targetSdkVersion="17" />  
  
  <application  
    android:allowBackup="true"  
    android:icon="@drawable/ic_launcher"  
    android:label="@string/app_name"  
    android:theme="@style/AppTheme" >  
    <activity  
      android:name="com.example.cancel_alarm_intent.MainActivity"  
      android:label="@string/app_name" >  
      <intent-filter>
```

```

        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

    <service
        android:name=".MyService"
        android:enabled="true" />
</application>

</manifest>

```

14.

### MainActivity.java

```

package com.example.service_service_regularly;

import java.util.Calendar;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity implements OnClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button start = (Button) findViewById(R.id.start);
        Button end = (Button) findViewById(R.id.end);
        start.setOnClickListener(this);
        end.setOnClickListener(this);
    }
}

```

```
}
```

```
@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}
```

```
@Override
```

```
public void onClick(View v) {  
    // TODO Auto-generated method stub  
    switch (v.getId()) {  
    case R.id.start:  
        startService(new Intent(this, MyService.class));  
        Calendar cal = Calendar.getInstance();  
        Intent intent = new Intent(this, MyService.class);  
        PendingIntent pintent = PendingIntent  
            .getService(this, 0, intent, 0);  
  
        AlarmManager alarm = (AlarmManager) getSystemService(Context.ALARM_SERVICE);  
        // Start service every hour  
        alarm.setRepeating(AlarmManager.RTC_WAKEUP, cal.getTimeInMillis(),  
            3600*1000, pintent);  
  
        break;  
    case R.id.end:  
        stopService(new Intent(this, MyService.class));  
        break;  
    default:  
        break;  
    }  
}
```

```
}
```

```
MyService.java
```

```
package com.example.service_service_regularly;
```

```
import android.app.Service;  
import android.content.Intent;  
import android.os.IBinder;  
import android.widget.Toast;
```

```
public class MyService extends Service {
```

```
    @Override
```

```
    public IBinder onBind(Intent intent) {  
        // TODO Auto-generated method stub  
        return null;  
    }
```

```
    @Override
```

```
    public void onCreate() {  
        Toast.makeText(this, " MyService Created ", Toast.LENGTH_LONG).show();  
    }
```

```
    @Override
```

```
    public void onStart(Intent intent, int startId) {  
        Toast.makeText(this, " MyService Started", Toast.LENGTH_LONG).show();  
    }
```

```
    @Override
```

```
    public void onDestroy() {  
        // TODO Auto-generated method stub  
        Toast.makeText(this, "Services Stopped", Toast.LENGTH_SHORT).show();  
        super.onDestroy();  
    }
```

```
}
```

```
Activity_main.xml
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    tools:context=".MainActivity" >
```

```
<Button
```

```
    android:id="@+id/end"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

```
android:layout_alignParentLeft="true"
android:layout_alignRight="@+id/start"
android:layout_below="@+id/start"
android:layout_marginTop="44dp"
android:text="End Service" />
```

### <Button

```
android:id="@+id/start"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true"
android:layout_marginTop="42dp"
android:text="Start Service" />
```

### </RelativeLayout>

### AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.service_service_regularly"
    android:versionCode="1"
    android:versionName="1.0" >
```

### <uses-sdk

```
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />
```

### <application

```
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
```

### <activity

```
    android:name="com.example.service_service_regularly.MainActivity"
    android:label="@string/app_name" >
```

### <intent-filter>

```
    <action android:name="android.intent.action.MAIN" />
```

```
    <category android:name="android.intent.category.LAUNCHER" />
```

### </intent-filter>

```
</activity>
```

```
<service
```

```
    android:name=".MyService"
```

```
    android:enabled="true" />
```

```
</application>
```

```
</manifest>
```

15.

### Main Activity

```
package com.example.texttospeech;
```

```
import java.util.Locale;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.speech.tts.TextToSpeech;
```

```
import android.util.Log;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
public class MainActivity extends Activity implements
```

```
    TextToSpeech.OnInitListener {
```

```
    private TextToSpeech tts;
```

```
    private Button btn;
```

```
    private EditText txt;
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    tts = new TextToSpeech(this, this);
```

```
    btn = (Button) findViewById(R.id.button1);
```

```
    txt = (EditText) findViewById(R.id.editText1);
```

```
    btn.setOnClickListener(new View.OnClickListener() {
```

```

    @Override
    public void onClick(View arg0) {
        speak();
    }

});
}

@Override
public void onDestroy() {
    if (tts != null) {
        tts.stop();
        tts.shutdown();
    }
    super.onDestroy();
}

@Override
public void onInit(int status) {
    if (status == TextToSpeech.SUCCESS) {
        int result = tts.setLanguage(Locale.US);
        if (result == TextToSpeech.LANG_MISSING_DATA
            || result == TextToSpeech.LANG_NOT_SUPPORTED) {
            Log.e("TTS", "This Language is not supported");
        } else {
            btn.setEnabled(true);
            speak();
        }
    } else {
        Log.e("TTS", "Initilization Failed!");
    }
}

private void speak() {
    String text = txt.getText().toString();
    tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}
}

```

16.

### **MainActivity.java**

```
package com.example.readwrite_file;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {

    private static final String TAG = MainActivity.class.getName();
    private static final String FILENAME = "myFile.txt";

    TextView text;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        text = (TextView) findViewById(R.id.textView1);

        Button Read = (Button) findViewById(R.id.read);
        Button write = (Button) findViewById(R.id.writefile);
```

```

Read.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        readFromFile();
        Toast.makeText(getApplicationContext(),
            "Successfully read From file", Toast.LENGTH_SHORT)
            .show();
    }
});
final String textToSaveString = "Hello Android";

```

```

write.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        writeToFile(textToSaveString);
        Toast.makeText(getApplicationContext(), "Write made to file",
            Toast.LENGTH_SHORT).show();

    }
});
}

```

```

private void writeToFile(String data) {
    try {

        FileOutputStream fos = openFileOutput(FILENAME,
            Context.MODE_PRIVATE);
        fos.write(data.getBytes());
        fos.close();

    } catch (IOException e) {
        Log.e(TAG, "File write failed: " + e.toString());
    }
}

```

```

private void readFromFile() {

```

```

File file = new File(this.getFilesDir() + "/" + FILENAME);
if (!file.exists()) {
    throw new RuntimeException("File not found");
}
Log.e("Testing", "Starting to read");
BufferedReader reader = null;
StringBuilder builder = null;
try {
    reader = new BufferedReader(new FileReader(file));
    builder = new StringBuilder();
    String line;
    while ((line = reader.readLine()) != null) {
        builder.append(line);
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (reader != null) {
        try {
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
text.setText(builder.toString());
}
}

```

### Activity\_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/writefile"

```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true"
android:layout_marginTop="35dp"
android:text="Write" />
```

#### <Button

```
android:id="@+id/read"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/writefile"
android:layout_alignParentRight="true"
android:layout_centerVertical="true"
android:text="Read" />
```

#### <TextView

```
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/read"
android:layout_alignParentBottom="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/read"
android:layout_marginTop="58dp"
android:text="Output Text"
android:textAppearance="?android:attr/textAppearanceLarge" />
```

#### </RelativeLayout>

17.

```
package com.example.hello_android_sanfoundry;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

```

```

}

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity"
>

```

```

<!-- This is a comment -->

```

```

<!-- Create your comments like this -->

```

```

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignParentTop="true"
    android:layout_marginTop="18dp"
    android:text="@string/hello_sanfoundry_android" />

```

```

<TextView

```

```
android:id="@+id/textView1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginTop="53dp"
android:text="This is a Large Text View"
android:textAppearance="?android:attr/textAppearanceLarge" />
```

#### <TextView

```
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_below="@+id/textView1"
android:layout_marginTop="16dp"
android:text="This is Medium Text View"
android:textAppearance="?android:attr/textAppearanceMedium" />
```

#### <TextView

```
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView3"
android:layout_below="@+id/textView3"
android:layout_marginTop="14dp"
android:text="Small Text"
android:textAppearance="?android:attr/textAppearanceSmall" />
```

#### <Button

```
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView4"
android:layout_below="@+id/textView4"
android:layout_marginTop="16dp"
android:text="Button" />
```

#### <ToggleButton

```
android:id="@+id/toggleButton1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:layout_alignBaseline="@+id/button2"  
android:layout_alignBottom="@+id/button2"  
android:layout_alignParentRight="true"  
android:text="ToggleButton" />
```

### **<Button**

```
android:id="@+id/button2"  
style="?android:attr/buttonStyleSmall"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignTop="@+id/button1"  
android:layout_centerHorizontal="true"  
android:text="Small Button" />
```

### **<CheckBox**

```
android:id="@+id/checkbox1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignRight="@+id/toggleButton1"  
android:layout_below="@+id/toggleButton1"  
android:layout_marginTop="93dp"  
android:text="CheckBox" />
```

### **<CheckedTextView**

```
android:id="@+id/checkedTextView1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/checkbox1"  
android:layout_alignBottom="@+id/checkbox1"  
android:layout_alignLeft="@+id/button1"  
android:text="CheckedTextView" />
```

### **<RadioButton**

```
android:id="@+id/radioButton1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/checkedTextView1"  
android:layout_below="@+id/checkbox1"  
android:layout_marginTop="34dp"  
android:text="RadioButton" />
```

### **<RadioButton**

```

    android:id="@+id/radioButton2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/radioButton1"
    android:layout_below="@+id/radioButton1"
    android:text="RadioButton" />

```

</RelativeLayout>

18.

### Main Activity

```
package com.example.string;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
}
```

### strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
    <string name="app_name">string</string>
```

```
    <string name="action_settings">Settings</string>
```

```
<string name="hello_world">Hello world!</string>
<string name="my_text">Definition of text!</string>
```

```
<string-array name="operating_systems">
  <item>Android</item>
  <item>iPhone</item>
  <item>Windows Mobile</item>
  <item>Linux </item>
  <item>Windows 7.0</item>
  <item>Windows 8.0</item>
  <item>Windows 8.1</item>
  <item>Chrome OS</item>
  <item>Unix</item>
  <item>Symbian</item>
</string-array>
```

```
</resources>
```

### **activity\_main**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  tools:context=".MainActivity" >
```

```
<TextView
  android:id="@+id/textView2"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/hello_world" />
```

```
<TextView
  android:id="@+id/textView1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignParentLeft="true"
  android:layout_alignParentRight="true"
  android:layout_below="@+id/textView2"
```

```

    android:layout_marginTop="110dp"
    android:text="@string/my_text"
    android:textAppearance="?android:attr/textAppearanceLarge" />

```

```
</RelativeLayout>
```

19.

```
package com.example.hello_android_sanfoundry;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
}
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    tools:context=".MainActivity"
```

```
    android:background="@drawable/background"
```

>

**<TextView**

```
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignParentTop="true"
    android:layout_marginTop="18dp"
    android:text="@string/hello_sanfoundry_android" />
```

**<TextView**

```
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="53dp"
    android:text="This is a Large Text View"
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

**<TextView**

```
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="16dp"
    android:text="This is Medium Text View"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

**<TextView**

```
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView3"
    android:layout_below="@+id/textView3"
    android:layout_marginTop="14dp"
    android:text="Small Text"
    android:textAppearance="?android:attr/textAppearanceSmall" />
```

**<Button**

```
    android:id="@+id/button1"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView4"
android:layout_below="@+id/textView4"
android:layout_marginTop="16dp"
android:text="Button" />
```

### <ToggleButton

```
android:id="@+id/toggleButton1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button2"
android:layout_alignBottom="@+id/button2"
android:layout_alignParentRight="true"
android:text="ToggleButton" />
```

### <Button

```
android:id="@+id/button2"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/button1"
android:layout_centerHorizontal="true"
android:text="Small Button" />
```

### <CheckBox

```
android:id="@+id/checkbox1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignRight="@+id/toggleButton1"
android:layout_below="@+id/toggleButton1"
android:layout_marginTop="93dp"
android:text="CheckBox" />
```

### <CheckedTextView

```
android:id="@+id/checkedTextView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/checkbox1"
android:layout_alignBottom="@+id/checkbox1"
android:layout_alignLeft="@+id/button1"
android:text="CheckedTextView" />
```

### <RadioButton

```
android:id="@+id/radioButton1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/checkedTextView1"  
android:layout_below="@+id/checkbox1"  
android:layout_marginTop="34dp"  
android:text="RadioButton" />
```

### <RadioButton

```
android:id="@+id/radioButton2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/radioButton1"  
android:layout_below="@+id/radioButton1"  
android:text="RadioButton" />
```

### </RelativeLayout>

МДК 01-04. Системное программирование  
Работа с файловой системой и процессами в Windows  
Примеры программ на языке Python.

#### 1. Создание и удаление директорий

```
import os
```

```
os.mkdir("Folder") # make directory  
os.chdir("Folder") # change (current) directory
```

```
for i in range(1, 51):  
    name = "Папка" + str(i)  
    os.mkdir(name)
```

```
for i in range(31, 41):  
    name = "Папка" + str(i)  
    os.rmdir(name)
```

## 2. Создание директорий и файлов

```
import os

os.chdir("Folder") # change (current) directory

for i in range(1, 21):
    name = "Папка" + str(i)
    os.chdir(name)
    f = open("Файл" + str(i) + ".txt", "w")
    f.write("Мой первый файл в папке номер " + str(i) + "\n")
    f.close()
    os.chdir("../")
```

## 3. Создание вложенных директорий

```
import os

lst1 = os.listdir()
print(lst1)

str1 = os.getcwd()
print(str1)

path = "C:\\Pycharm_projects\\GeeksForGeeks\\Authors"

# Create the directory
os.makedirs(path)

print(os.getpid())
```

## 4. Запуск других программ на Python в исходной программе и возврат управления к ней

```
import os, sys
import subprocess

N = 5
for i in range(1, N+1):
    name = "Папка" + str(i)
    os.mkdir(name)
```

```
s1 = os.getcwd()+"\\potomok1.py"
subprocess.call(["python", s1])
```

```
for i in range(1, N+1):
```

```
    name = "Папка" + str(i)
```

```
    os.chdir(name)
```

```
    subprocess.call(["python", s1, str(i)]) # передать потомку параметр i через argv чтобы он создал
файл с номером i
```

```
    os.chdir("../")
```

#### *5. Вспомогательная программа для прошлого примера*

```
import os, sys
```

```
print('Hello from child', os.getpid(), sys.argv)
```

#### *6. Отобразить постранично содержимое файла, указанного в командной строке*

```
def more(text, numlines=15):
```

```
    lines = text.splitlines() # подобно split('\n') но без '' в конце
```

```
    while lines:
```

```
        chunk = lines[:numlines]
```

```
        lines = lines[numlines:]
```

```
        for line in chunk: print(line)
```

```
        if lines and input('More?') not in ['y', 'Y']: break
```

```
import sys
```

```
more(open(sys.argv[1]).read(), 10)
```

#### *7. Работа с файлами и с аргументами командной строки*

```
import sys
```

```
f = open("File.txt", "w+")
```

```
if len(sys.argv)==1:
```

```
    f.write("My first file\n")
```

```
else:
```

```
    f.write("My first file in Folder" + sys.argv[1] + "\n")
```

```
f.close()
```

```
if len(sys.argv)==1:  
    print("Hello from potomok")  
else:  
    print("Hello from potomok in Folder" + sys.argv[1])
```

#### 8. Работа с файлами и случайными числами

```
f = open("randomfail.txt", "w+")  
import random  
for i in range (1000):  
    x= random.randint(0, 1000)  
    f.write(str(x)+ " ")  
    if i%10== 0 : f.write("\n")  
f.close()
```

#### 9. Принудительное завершение дочерних процессов программы

```
# example of killing a child process by pid  
from time import sleep  
from multiprocessing import Process  
from os import kill  
from os import getpid  
from signal import SIGABRT  
  
# function executed in a child process  
def task():  
    # report a message  
    print('Child is running...', flush=True)  
    # block for a while  
    sleep(20)  
  
# protect the entry point  
if __name__ == '__main__':  
    # start a child process  
    child = Process(target=task)  
    child.start()  
    # wait a moment  
    sleep(1)  
    # report a message
```

```
print(f'Child is running with pid: {child.pid}')
# attempt to kill the child process
kill(child.pid, SIGABRT)
# wait for child to terminate
child.join()
# report status of the child
print(f'Child process: {child}')
```

Изучение модуля asyncio для управления процессами (асинхронная, параллельная, последовательная работа процессов)

Примеры программ на языке Python. Нужно объяснить результаты выполнения данных программ.

1.

```
import asyncio
```

```
async def compute(x, y):
    print("Compute %s + %s ..." % (x, y))
    await asyncio.sleep(1.0)
    return x + y
```

```
async def print_sum(x, y):
    result = await compute(x, y)
    print("%s + %s = %s" % (x, y, result))
```

```
loop = asyncio.get_event_loop()
loop.run_until_complete(print_sum(1, 2))
loop.close()
```

2.

```
import asyncio
async def fn():
    task=asyncio.create_task(fn2())
    print(1)
    print(2)
    await asyncio.sleep(5)
    print(4)
```

```
await asyncio.sleep(5)
```

```
async def fn2():  
    print(3)  
    await asyncio.sleep(5)  
    print(5)
```

```
asyncio.run(fn())
```

3.

```
import asyncio
```

```
async def func1():  
    print("Function 1 started..")  
    await asyncio.sleep(2)  
    print("Function 1 Ended")
```

```
async def func2():  
    print("Function 2 started..")  
    await asyncio.sleep(3)  
    print("Function 2 Ended")
```

```
async def func3():  
    print("Function 3 started..")  
    await asyncio.sleep(1)  
    print("Function 3 Ended")
```

```
async def main():  
    await asyncio.gather(  
        func1(),  
        func2(),  
        func3(),  
    )  
    print("Main Ended..")
```

```
asyncio.run(main())
```

4.

```
import asyncio

async def foo():
    print("Start foo")
    await asyncio.sleep(1)
    print("End foo")

async def bar():
    print("Start bar")
    await asyncio.sleep(2)
    print("End bar")

async def main():
    await asyncio.gather(foo(), bar())

asyncio.run(main())
```

5.

```
import asyncio

async def foo(delay):
    for i in range(10):
        print(i)
        await asyncio.sleep(delay)

def stopper(loop):
    loop.stop()

loop = asyncio.get_event_loop()

# Schedule a call to foo()
loop.create_task(foo(0.5))
loop.create_task(foo(1))
loop.call_later(12, stopper, loop)

# Block until loop.stop() is called()
```

```
loop.run_forever()
loop.close()
```

6.

```
import asyncio

async def count_to_three():
    for i in range(1, 4):
        print(i)
        await asyncio.sleep(1)

async def main():
    await asyncio.gather(count_to_three(), count_to_three())

# Python 3.7+
asyncio.run(main())
```

7.

```
import asyncio
import tkinter
from tkinter import messagebox
import time

def fn():
    fn2()
    print(1)
    time.sleep(5)
    print(4)
    time.sleep(1)

def fn2():
    #await asyncio.sleep(1)
    win = tkinter.Tk()
    messagebox.showinfo('For testing', 'Some text...')
    win.destroy()
    time.sleep(1)
    print("fn2 here")
```

```
fn()
```

8.

```
import asyncio
import time

async def async_job(number):
    print(f'Job {number} started')
    await asyncio.sleep(1)
    print(f'Job {number} finished')

def sync_job(number):
    print(f'Job {number} started')
    time.sleep(1)
    print(f'Job {number} finished')

# Асинхронный вызов
asyncio.run(async_job(1))

# Синхронный вызов
sync_job(2)
```

9.

```
import asyncio
import tkinter
from tkinter import messagebox

async def fn():
    task=asyncio.create_task(fn2())
    print(1)
    await asyncio.sleep(5)
    print(4)
    await asyncio.sleep(1)

async def fn2():
    #await asyncio.sleep(1)
    win = tkinter.Tk()
    messagebox.showinfo('For testing', 'Some text...')
```

```
win.destroy()
await asyncio.sleep(1)
print("fn2 here")
```

```
asyncio.run(fn())
```

Следующие задания необходимо решить, написав программу на языке Python. Обычно в них бывают нужны модули `os` и `sys` из языка Python.

Если речь идет о каком-либо файле, его нужно заранее подготовить в простом текстовом редакторе и сохранять с расширением `.txt`

Подобные задания также давались на диф. зачете.

1. Создайте программу для вывода содержимого файла в окно консоли.

а) Имя файла запрашивается при запуске программы и вводится с клавиатуры

б) Имя файла задается в командной строке:

<имя программы на Python> <имя файла для этой программы>

2. В файле записаны целые числа. Создайте программу для вывода этих чисел по  $N$  чисел в каждой строке (число  $N$  вводится с клавиатуры ( $N > 0$ ) и имеет значение по умолчанию = 10). В случае, если файл пустой, выведите сообщение «пустой файл».

а) Имя файла запрашивается при запуске программы и вводится с клавиатуры

б) Имя файла задается в командной строке:

<имя программы на Python> <имя файла для этой программы>

3. а) Напишите программу, которая будет открывать определенный файл, введенный с клавиатуры, и выводить на печать построчно последние строки в количестве  $N$  (которое тоже вводится с клавиатуры и должно быть положительным).

б) То же самое для случая, когда имя файла задается в командной строке:

<имя программы на Python> <имя файла для этой программы>

4. Дан текстовый файл. Записать в новый файл все слова из первого файла, начинающиеся с заданной буквы. Словом считать последовательность букв, ограниченных пробелами или знаками препинания. Имя файла задается с клавиатуры, и заданная буква - тоже.

5. Создать файл, содержащий заданное количество  $N$  случайных целых чисел в диапазоне  $[-1000..1000]$ , а также вывести числа на экран в обратном порядке и записать в другой файл в обратном порядке. Имя файла задается с клавиатуры, число  $N$  – тоже.

6. Даны два файла с целыми числами. Сравнить их и записать в третий файл числа, которые есть во втором файле, но не встречались в первом. Имена файлов задать с клавиатуры.

7. Дан файл под названием "Автовладельцы". В нём содержатся автовладельцы с информацией: номер, марка авто, фамилия и адрес (для каждого автовладельца отдельная строка). Написать программу, которая выводит на экран и записывает в отдельный файл тех автовладельцев, у которых более одного автомобиля.
8. Имеется текстовый файл, состоящий из N строк, записанных в столбик. Все четные строки этого файла записать во второй файл, а нечетные — в третий файл. Порядок следования строк сохраняется. Имена файлов задаются с клавиатуры.
9. Дан файл. Создать новый файл, полученный из первого файла заменой всех его прописных букв соответствующими строчными. Имена файлов задаются с клавиатуры.
10. Имеется текстовый файл, состоящий из строк, записанных в столбик. Переписать его строки в другой файл. Порядок строк во втором файле должен быть обратным по отношению к порядку строк в заданном файле.  
Имена файлов задаются с клавиатуры.
11. Напишите программу, которая принимает поисковый запрос и выводит названия текстовых файлов, содержащих искомую подстроку. Все файлы располагаются в текущей директории.
12. В заданном текстовом файле на английском языке подсчитать частоту вхождения каждой буквы. Строчные и прописные буквы не различаются, все остальные символы пропускаются. Имя файла задавать с клавиатуры.
13. Создать текстовый файл, заполненный случайными N целыми числами, записанными через пробел. Вывести в файл в отдельные строки сумму этих чисел, среднее арифметическое и произведение.
14. В файле содержится совокупность текстовых строк. Изменить первую букву каждого слова на заглавную. Имя файла задавать с клавиатуры.
15. В данном текстовом файле удалить все слова, которые содержат хотя бы одну цифру. Имя файла задавать с клавиатуры.
16. Создать и заполнить файл случайными целыми значениями. Выполнить сортировку содержимого файла по возрастанию. Имя файла задавать с клавиатуры.
17. Получить файл, в котором текст выровнен по правому краю путем равномерного добавления пробелов. Имя исходного файла задавать с клавиатуры.
18. Дан файл, содержащий различные даты. Каждая дата – это число, месяц и год в формате dd:mm:yy . Найти самую раннюю дату (вывести на экран). Имя исходного файла задавать с клавиатуры.
19. Дан файл, компоненты которого являются целыми числами. Найти количество удвоенных нечётных чисел среди компонент (вывести на экран). Имя исходного файла задавать с клавиатуры.
20. Дан файл, содержащий различные даты. Каждая дата – это число, месяц и год в формате dd:mm:yy . Найти все весенние даты (вывести на экран). Имя исходного файла задавать с клавиатуры.

#### 4.4. Типовые задания для оценки освоения МДК.01.04 Системное программирование

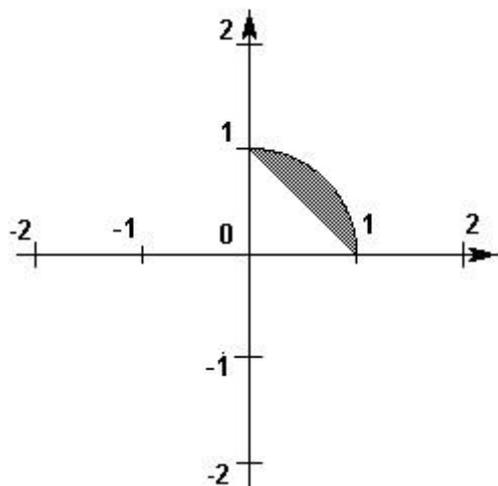
##### Вопросы для подготовки к экзамену/зачёту 1.

Что такое системное программирование?

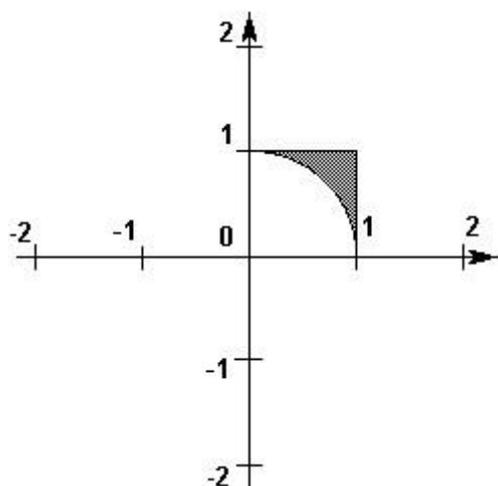
2. Инструментальные средства для создания и запуска программ на языке C
3. Структура программы на языке C
4. Интерпретаторы и компиляторы
5. Программирование на языке C в среде Visual Studio. Отладка программы
6. Алфавит языка C
7. Идентификаторы
8. Классификация данных в языке C
9. Типы данных языка C
10. Объявление и инициализация переменных в языке C
11. Ввод данных с клавиатуры и вывод данных на экран в языке C
12. Работа с файлами в языке C. Чтение данных из файла и запись в файл
13. Классификация операций языка C. Основные операции
14. Логические переменные в языке C. Операции отношения, сравнения, логические операции, поразрядные логические операции, операции сдвига, условная операция
15. Математические функции языка C
16. Видимость переменных в языке C, локальные и глобальные переменные, статические переменные
17. Условный оператор
18. Оператор выбора
19. Циклические программы. Классификация циклов
20. Цикл с предусловием
21. Цикл с постусловием
22. Цикл с заданным числом повторений
23. Алгоритм вычисления суммы бесконечного ряда с заданной точностью
24. Алгоритм вычисления определенного интеграла с заданной точностью
25. Алгоритм решения уравнений методом дихотомии
26. Функции в языке C. Описание, вызов функции, прототипы функций
27. Передача параметров в функции по имени и по адресу
28. Рекурсивные функции
29. Создание многомодульных программных проектов
30. Массивы. Описание массивов и их инициализация в языке C. Доступ к элементам массива
31. Методы сортировки одномерных массивов
32. Алгоритмы нахождения минимального (максимального) элемента массива и подсчёта суммы элементов массива

### Примеры практических заданий

1. Написать программу, которая удаляет из введенной с клавиатуры строки первые пробелы (если их нет, то оставить как есть)
2. Подсчитать количество слов в строке
3. Выделить первое слово из строки
4. Найти максимальное из трех введенных целых чисел
5. Найти сумму цифр трехзначного числа
6. Решить квадратное уравнение  $ax^2+bx+c=0$ , коэффициенты уравнения вводятся с клавиатуры
7. Определить, кратна ли трем сумма цифр трехзначного числа. Число вводится с клавиатуры
8. Найти площадь прямоугольного треугольника по известным катету и гипотенузе. Исходные данные вводятся с клавиатуры
9. Переписать все символы строки в обратном порядке
10. Написать программу, которая вводит координаты точки  $(x, y)$  и определяет, попадает ли точка в заштрихованную область на рисунке. Попадание на границу области считать попаданием в область.



11. Построить программу, которая вводит координаты точки  $(x, y)$  и определяет, попадает ли точка в заштрихованную область на рисунке. Попадание на границу области считать попаданием в область.



12. Написать программу для сортировки одномерных массивов
13. Вывести на экран все трехзначные числа, в которых хотя бы две цифры одинаковые
14. Вывести на экран все трехзначные числа, которые начинаются и заканчиваются на одну и ту же цифру
15. Вывести на экран таблицу значений функции  $y=x^3$  при  $x=-4\dots4$ . Шаг изменения  $x$  вводится с клавиатуры
16. Найти сумму ряда с точностью 0.0001

$$a_n = (-1)^n \frac{n+1}{n^3+2}$$

17. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 найти сумму положительных элементов
18. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 количество отрицательных элементов
19. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 найти среднее арифметическое четных элементов
20. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 заменить все отрицательные элементы нулями
21. В одномерном массиве, состоящем из 10 случайных чисел от -10 до 10 переставить местами минимальный и максимальный элементы
22. Определить среднее арифметическое элементов главной диагонали квадратной матрицы
23. Определить максимальный элемент побочной диагонали квадратной матрицы 24. Найти сумму нечетных элементов прямоугольного массива, состоящего из пяти строк и семи столбцов
25. В прямоугольном массиве, состоящем из 7 строк и 10 столбцов заменить все отрицательные элементы их абсолютными значениями
26. Определить, является ли сумма элементов матрицы 3x2 нечетной
27. В прямоугольном массиве, состоящем из 7 строк и 10 столбцов заменить все отрицательные элементы их абсолютными значениями
28. Определить, является ли сумма элементов матрицы 3x2 нечетной
29. В прямоугольном массиве, состоящем из 7 строк и 10 столбцов заменить все отрицательные элементы их абсолютными значениями

30. Определить, является ли сумма элементов матрицы  $3 \times 2$  нечетной

#### **4.5. Типовые задания для экзамена квалификационного по ПМ.01**

##### **Задание 1**

Разработать приложение с пользовательским графическим интерфейсом по предложенному заданию

##### **Задание 2**

Провести отладку и тестирования разработанного приложения **Задание**

##### **3**

Создать мобильное приложение, реализующее часть функционала созданного приложения

## 5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	<p>Оценка <b>«отлично»</b> - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры. указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма</p> <p>Оценка <b>«хорошо»</b> - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры. <b>выполнена оценка сложности алгоритма</b></p> <p>Оценка <b>«удовлетворительно»</b> - алгоритм разработан и соответствует заданию.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p>

<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
--	--	--

	<p>программирования методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	
--	---	--

<p>ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Оценка <b>«отлично»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«хорошо»</b> - выполнена отладка модуля с использованием инструментария среды проектирования; сохранены и представлены результаты отладки.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.4 Выполнять тестирование программных модулей</p>	<p>Оценка <b>«отлично»</b> - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами. Оценка <b>«хорошо»</b> - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования.</p> <p>Оценка <b>«удовлетворительно»</b> - выполнено тестирование модуля и оформлены результаты тестирования.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля.</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Оценка <b>«отлично»</b> - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены</p>	<p>Экзамен/зачет в форме собеседования: практическое</p>

	<p>фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p> <p>Оценка «<b>хорошо</b>» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p> <p>Оценка «<b>удовлетворительно</b>» - определены качественные характеристики программного кода частично с помощью инструментальных средств; выявлено несколько фрагментов некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p>	<p>задание по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
--	--	--

<p>ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.</p>	<p>Оценка <b>«отлично»</b> - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации.</p> <p>Оценка <b>«хорошо»</b> - разработан модуль для заданного мобильного устройства с учетом основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие выполняемых функций спецификации с незначительными отклонениями.</p> <p>Оценка <b>«удовлетворительно»</b> - разработан модуль для заданного мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по созданию модуля для заданного мобильного устройства на основе спецификации</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
---	--	---

	<p>функций спецификации.</p>	
<p>ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p>	<p>П обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач;</p> <p>- адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач</p>	<p>Экспертное наблюдение за выполнением работ</p>
<p>ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p>	<p>- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернетресурсы, периодические издания по специальности для решения профессиональных задач</p>	
<p>ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<p>- демонстрация ответственности за принятые решения</p> <p>- обоснованность самоанализа и коррекция</p>	

	результатов собственной работы;	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	- взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей	
ОК 06. Проявлять гражданскопатриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.	- соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик,	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	- эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	

<p>ОК 09. Использовать информационные технологии в профессиональной деятельности.</p>	<p>- эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;</p>
<p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.</p>	<p>- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.</p>
<p>ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере</p>	<p>- эффективность использования в профессиональной деятельности предпринимательских навыков</p>

ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 133600552358087161194895262509558337786447861787

Владелец Гайдаров Насир Алиевич

Действителен с 22.03.2024 по 22.03.2025