Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 06.09.2023 16:41:39
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета

«Информационные технологии»

_____/Д.Г.Демидов/

«18» мая 2022

Рабочая программа дисциплины

«Объектно-ориентированное проектирование»

Направление подготовки:

09.03.03 Прикладная информатика

Образовательная программа (профиль):

«Корпоративные информационные системы»

Год начала обучения:

2022

Уровень образования:

Бакалавриат

Квалификация (степень) выпускника:

Бакалавр

Форма обучения:

очная

Москва, 2022

Рабочая программа дисциплины «Объектно-ориентированное проектирование» составлена в соответствии с федеральным образовательным стандартом высшего образования – бакалавриата по направлению подготовки 09.03.03 Прикладная информатика.

Заведующий кафедрой «Инфокогнитивные технологии»:



к.т.н., доцент /Е.А.Пухова /

Согласовано:

Руководитель образовательной программы:



_____ /М.В.Даньшина/

Программу составили:

_____ / М.В.Даньшина/

_____ / _____ /

_____ / _____ /

_____ / _____ /

_____ / _____ /

1. Цели освоения дисциплины

К **основным целям** освоения дисциплины относится:

изучение методов объектно-ориентированного анализа и проектирования;

формирование навыков использования современных технологий программирования;

изучение приемов разработки программных приложений,

ориентированных на повторный код (методы повторного использования);

изучение унифицированного языка моделирования UML;

закрепление получаемых в семестре знаний и навыков на практике; формирование взаимосвязей, получаемых в семестре знаний и

навыков с изученными ранее и изучаемых параллельно с данной дисциплиной; подготовка студентов к деятельности в соответствии с

квалификационной характеристикой бакалавра.

К **основным задачам** дисциплины относятся:

сформировать целостное представление о методах и подходах объектно-ориентированного анализа и проектирования программно-аппаратных комплексов, помочь овладеть практическим опытом программирования на языке C++ Java и проектирования программных продуктов с применением унифицированного языка моделирования UML (Unified Modeling Language) в среде Rational Rose;

программирование приложений, создание прототипа информационной системы, документирование проектов информационной системы на стадиях жизненного цикла, использование функциональных и технологических стандартов;

изучение и освоение теоретического материала, как в процессе контактной, так и в ходе самостоятельной работы;

выполнение предоставленных практических заданий различных форм, как в процессе контактной, так и в ходе самостоятельной работы;

самостоятельная работа над тематикой дисциплины для формирования компетенций основной образовательной программы (далее, ООП).

2. Место дисциплины в структуре ООП

Дисциплина относится к числу учебных обязательных дисциплин основной образовательной программы.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

Алгоритмизация и программирование;

Основы разработки КИС;

Разработка КИС;

Инженерное проектирование;

Проектная деятельность.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций.

Код компетенции	Наименование компетенции	Индикаторы планируемых результатов обучения по дисциплине
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение	ИПК-3.1. Знать: методологии разработки программного обеспечения и технологии программирования; методологии и технологии проектирования и использования баз данных; языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования программных интерфейсов;

		методы и средства проектирования баз данных; принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения. ИПК-3.2. Уметь: выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ИПК-3.3. Владеть: современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.
ПК-4	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов	ИПК-4.1. Знать: методы описания алгоритмов, основные абстрактные типы данных и их реализации; проводить работы по интеграции программных модулей с использованием языков программирования.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом

и календарным графиком учебного процесса.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, т.е. 144 академических часов (из них 90 часов – самостоятельная работа студентов).

Разделы дисциплины изучаются на втором курсе в четвертом семестре, форма промежуточной аттестации - экзамен.

Содержание и темы лекций

Данный курс делает акцент не на синтаксисе самих языков и особенностей применения тех или иных конструкций, а на самой технологии объектно-ориентированного подхода и средствах поддержки именно принципов ООП в инструментальных языках. Изложение теории ООП ведется по схеме: Эволюция и принципы – Объект – Класс – Модуль – Система. Изложение средств языков по схеме:

- Для C++: Отличия от C – Объекты (классы, типизация) - Средства управления жизненным циклом объектов – Реализация отношений между объектами средствами классов – Средства построения иерархий классов – Средства построения программных модулей – Обобщенное программирование – Библиотека.

- Для Java: Платформа и структура машины – Объекты – Управление жизненным циклом – Реализация отношений между объектами (классами) – Иерархии классов – Модули – Библиотека – Параллельное программирование– Безопасность – Распределенное программирование.

Темы:

1. Сущность объектно-ориентированного подхода к анализу и проектированию ИС. Проблема сложности больших систем. Примеры больших систем. Сущность объектно-ориентированного подхода.

2. Взаимосвязь структурного подхода к проектированию информационных систем. История развития проектирования информационных систем. Сущность структурного и объектно-ориентированного подходов. Сущность объектно-ориентированного подхода к проектированию информационных систем. Отличие между структурным и объектно-ориентированным подходом к проектированию программных комплексов.

3. Унифицированный язык моделирования. Введение. Этапы и принципы разработки сложных систем. Объектно-ориентированный анализ. Язык UML. Паттерны проектирования.

4. Инфологическое проектирование. Диаграмма прецедентов.

5. Диаграмма последовательностей.

6. Логическое проектирование. Диаграмма классов.

7. Диаграмма кооперации.

8. Физическое проектирование. Диаграмма реализации. Проверка модели. Кодогенерация.

9. Диаграмма состояний.

Содержание и темы лабораторных работ

ЛР-1	**Типы данных, операции и условные операторы языка программирования Java**	8 ак. часов
Цель выполнения лабораторной работы: знакомство со структурой языка программирования Java.		
Результат: Работающая программа согласно полученному заданию		

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

приобретение навыков: применения арифметических и логических операций; работы с простыми типами данных; использования условных операторов; использования класса Math для различных математических операций.

Выполнение задания по индивидуальному заданию.

Защита лабораторной работы.

Контрольные вопросы:

1. Что такое Java?
2. Какой основное назначение Java?
3. В чем отличие статического и динамического контента?
4. Где выполняется Java -код?
5. Какие математические операторы доступны в Java?

ЛР-2	**Циклические алгоритмы. Условия в алгоритмах.**	8 ак. часов

Цель выполнения лабораторной работы: Изучение структуры и синтаксиса языка Java, получение базовых навыков построения программ, использования основных операторов. Закрепление знаний использования циклов и условных операторов, как основы построения алгоритмов.

Результат: Работающая программа согласно полученному заданию

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

изучение математических операторов Java; изучение условных операторов Java; изучение операторов цикла в Java. Получение варианта задания.

Разработка программы, согласно требованиям.

Тестирование программы, в том числе на устойчивость работы при некорректных данных.

Защита лабораторной работы.

Контрольные вопросы:

1. Что такое переменная?
2. Что такое цикл?
3. Что такое условный оператор?
4. Какие виды циклов бывают?
5. Чем цикл с предусловием отличается от цикла с постусловием?
6. Чем отличаются операторы break и continue?
7. Что такое итерация?
8. Какой тип данных определяет условие?
9. Что такое булева алгебра?
10. Какие операции булевой алгебры (логические операции) Вы знаете?
11. Как в Java округлить значение до заданной точности?
12. Как в Java преобразовать вещественный тип к целому?
13. Какие виды округления используются в Java?
14. Что такое конструкция выбора?
15. Какие правила преобразования типов существуют в Java?

ЛР-3	**Работа со строками в языке программирования Java.**	4 ак. часа

Цель выполнения лабораторной работы: Знакомство с типами данных String и StringBuffer, получение навыков создания и обработки строк

Результат: Работающая программа согласно полученному заданию

Порядок выполнения лабораторной работы:

Подготовка к выполнению лабораторной работы – приобретение навыков работы со строками в языке программирования Kotlin, применения следующих операций: создание и инициализация объекта класса String; создание и инициализация объекта класса StringBuffer; определение размера строки; модификация строк; сравнение строк; поиск в строках; извлечение символов и подстрок из строки.

Защита лабораторной работы.

Контрольные вопросы:

1. Что такое String в Koltin? Какой это тип данных?
2. Какие есть способы создания объекта String?
3. Напишите метод проверки, является ли строка палиндромом.
4. Как мы можем перевести строку в верхний регистр или нижний регистр?
5. В чем отличие String от StringBuffer?

ЛР-4	**Объектно-ориентированное программирование. Инкапсуляция, полиморфизм и наследование.**	8 ак. часов

Цель выполнения лабораторной работы: Знакомство с объектно-ориентированным программированием, получение навыков представления объектов реального мира из какой-либо предметной области в форме классов с набором полей и методов на языке программированияKotlin.

Результат: Работающая программа согласно полученному заданию

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

приобретение навыков создания собственных классов и объектов на языке программирования Java, а также изучить основные конструкции, используемые при построении классов и объектов; разобраться с понятием инкапсуляция; разобраться с понятием полиморфизм; разобраться с понятием наследование; изучить основные конструкции, используемые при создании интерфейсов и их реализация;

создание базового класса по варианту.

создание подкласса созданного базового класса.

создание иерархии классов.

создание интерфейса по варианту.

реализация созданного интерфейса в базовом классе

Защита лабораторной работы.

Контрольные вопросы:

1. Назовите принципы ООП и расскажите о каждом?
2. Дайте определение понятию “класс”?
3. Что такое поле/атрибут класса?
4. Как правильно организовать доступ к полям класса?
5. Дайте определение понятию “конструктор”?
6. Сколько раз инструкция return может быть использована в теле функции?
7. Сколько аргументов может быть передано функции?
8. Что-такое аргументы "по умолчанию"?
9. Может ли функция вообще не иметь аргументов?
10. Должны ли совпадать имена переменных-аргументов при объявлении и при вызове функции?

ЛР-5	**Создание Activity. Жизненный цикл Activity**	8 ак. часов

Цель выполнения лабораторной работы: Знакомство с пользовательским интерфейсом Android приложения

Результат: Apk-файл Android приложения

Порядок выполнения лабораторной работы:

Подготовка к выполнению лабораторной работы: изучить структуру проекта; разобраться с понятием Activity; разобраться в каких состояниях может быть Activity.

Создание проекта в Android Studio. Создание двух Activity.

Организация переходов между Activity.

Организация передачи параметров между Activity.

Защита лабораторной работы.

Контрольные вопросы:

1. Что такое Activity?
2. Что делает метод onCreate?
3. Расскажите про жизненный цикл Activity?
4. Как создать Activity?
5. Как передать параметр меджу Activity?
6. Как сделать Activity стартовой?
7. Что такое Intent?

ЛР-6	**Основы верстки Android приложений**	4 ак. часа

Цель выполнения лабораторной работы: Изучить основы создания пользовательского интерфейса Android приложений

Результат: Apk-файл Android приложения

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

изучить основы верстки; научиться управлять пользовательским интерфейсом мобильного приложения;

Создание проекта в Android Studio.

Создание трех Activity с пользовательским интерфейсом по варианту. Организация переходов между Activity.

Защита лабораторной работы.

Контрольные вопросы:

1. Что такое XML?
2. Какие типы ViewGroup бывают?
3. В чем отличие LinearLayout- и RelativeLayout?
4. Какие элементы формы существуют?
5. Как передает свое значение каждый элемент формы в Мобильное приложение?
6. Как перейти на другое Activity?

ЛР-7	**Создание мобильного приложения с использованием ListView и SharedPreferences**	12 ак. часов

Цель выполнения лабораторной работы: Приобретение навыков построения многооконного пользовательского интерфейса, хранение данных во внутренней памяти устройства

Результат: Мобильное приложение «Каталог фильмов». Apk-файл Android приложения.

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

изучить создание пользовательского интерфейса со списком элементов;

приобретение навыков хранения данных во внутренней памяти устройства. Создание проекта в Android Studio.

Создание экрана для авторизации в приложении.

Сохранение идентификатора входа в приложении.

Создание экрана со списком фильмов.

Организация перехода при нажатии на конкретный фильм из списка на новый экран, с передачей данных о фильме.

Создание меню в приложении для перехода между экранами. Создание функции выхода из приложения.

Создание экрана «О приложении»

Защита лабораторной работы.

Контрольные вопросы:

1. Как использовать SharedPreferences в Android для хранения, извлечения и редактирования значений?
2. Как удалить общие настройки?
3. Какая разница между commit() и apply() в общих предпочтениях?
4. Как получить SharedPreferences из PreferenceActivity в Android?
5. В чем разница между getDefaultSharedPreferences и getSharedPreferences в Android?
6. Как Android SharedPreferences сохраняет/сохраняет объект?
7. Где хранятся общие настройки?
8. Должен ли доступ к SharedPreferences делать из потока пользовательского интерфейса?

ЛР-8	**Современные методы организации сетевого взаимодействия Архитектура REST**	4 ак. часа

Цель выполнения лабораторной работы: Получение представлений об организации сетевого взаимодействия в Koltin, принципов работы

Результат: Apk-файл Android приложения

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

изучение способов построение сетевого взаимодействие;

изучение приемов работы REST;

Разработка программы для получения информации от сервера. Защита лабораторной работы.

Контрольные вопросы:

1. Что такое REST?
2. Какие способы организации сетевого взаимодействия вы знаете?
3. Можно ли сделать запрос в main потоке?
4. Что такое сериализации?
5. Как обработать ответ от сервера?

ЛР-9	**Создание мобильного приложения с использованием фрагментов.**	4 ак. часа

Цель выполнения лабораторной работы: Приобретение навыков построения пользовательского интерфейса с использованием фрагментов.

Результат: Мобильной приложение «Каталог фильмов и сериалов».Apk-файл Android приложения.

Порядок выполнения лабораторной работы:

Подготовка к выполнению к работе, в том числе:

приобретение навыков создания пользовательского интерфейса с использованием фрагментов;

приобретение навыков создания навигации в приложении при помощи BottomNavigationView.

Создание проекта в Android Studio.

Создание экрана для авторизации в приложении.

Сохранение идентификатора входа в приложении.

Создание экрана для хранения фрагментов

Создание фрагмента со списком фильмов.

Организация перехода при нажатии на конкретный фильм из списка на новый экран, с передачей данных о фильме.

Создание фрагмента со списком сериалов.

Создание меню в приложении для сортировки фильмов/сериалов. Создание функции выхода из приложения.

Создание экрана «О приложении»

Защита лабораторной работы.

Контрольные вопросы:

1. Что такое BottomNavigationView?
2. Что такое DDMS? Опишите некоторые из его возможностей?
3. В чем разница между фрагментом и активностью? Объясните взаимосвязь между ними?
4. В чем разница между Serializable и Parcelable? Что предпочтительнее использовать в Android?
5. В чем разница между Service и IntentService? Как они используются?

ЛР-10	**Создание мобильного приложения с использованием SQLite**	12 ак. часа

Цель выполнения лабораторной работы: Приобретение навыков работы с базой данных в мобильном приложении

Результат: Мобильной приложение «Каталог фильмов и сериалов».Apk-файл Android приложения.

Порядок выполнения лабораторной работы:

Создание проекта в Android Studio.

Создание экрана для авторизации в приложении.

Создание экрана для регистрации в приложении.

Сохранение идентификатора входа в приложении.

Создание экрана для хранения фрагментов

Создание фрагмента со списком фильмов.

Организация перехода при нажатии на конкретный фильм из списка на новый экран, с передачей данных о фильме.

Создание фрагмента со списком сериалов.

Создание фрагмента для отображения профиля пользователя.

Создание меню в приложении для сортировки фильмов/сериалов. Создание функции выхода из приложения.

Создание экрана «О приложении»

Защита лабораторной работы.

5. Образовательные технологии

Методика преподавания дисциплины и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование

следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков у обучающихся:

выполнение лабораторных работ в лабораториях вуза;

индивидуальные и групповые консультации студентов преподавателем, в том числе в виде защиты выполненных заданий в рамках самостоятельной работы;

посещение профильных конференций и работа на мастер-классах экспертов и специалистов индустрии.

Самостоятельная внеаудиторная работа студентов состоит из:

подготовки к выполнению и подготовки к защите лабораторных работ;

чтения литературы и освоения дополнительного материала в рамках тематики дисциплины;

подготовки к текущей аттестации;

подготовки к промежуточной аттестации.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов.

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

выполнение лабораторных работ, экзамен.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Объектно-ориентированное проектирование»

Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение
ПК-4	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплины в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.1. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины, описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	Критерии оценивания			
	2	3	4	5
ПК-3. Способен разрабатывать требования и проектировать программное обеспечение				
ИПК-3.1. Знать: методологии разработки программного обеспечения и технологии программирования;	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций

методологии и технологии проектирования и использования баз данных; языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования программных интерфейсов; методы и средства проектирования баз данных; принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения. ИПК-3.2. Уметь: выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения;	знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ИПК-3.3. Владеть: современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.				
ПК-4. Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов				
ИПК-4.1. Знать: методы описания алгоритмов, основные абстрактные типы данных и их реализации; проводить работы по интеграции программных модулей с использованием языков программирования.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

		испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.		

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине.

Шкала оценивания	**Описание**
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затрудненияпри
	аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений,

	навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература

1. Мухортов В.В., Рылов В.Ю. Объектно-ориентированное программирование, анализ и дизайн. Методическое пособие. ИМ СО РАН, 2002

2. Б. Страуструп Дизайн и эволюция языка C++, Пер. с англ. - ДМК Пресс, Питер, 2006г.

3. Брюс, Эккель, Философия Java, Пер. с англ. - Питер, 2003, 2009

4. Герберт Шилдт, Полный справочник по Java, Java SE 6th edition, 7-е издание, Пер. с англ. - Вильямс, 2007

5. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. — М. ДМК, 2000.

б) дополнительная литература:

1. Скотт Мейерс, Эффективное использование C++. 50 рекомендаций по улучшению ваших программ и проектов, Пер. с англ. - ДМК, 2006

2. Скотт Мейерс, Эффективное использование C++. 35 новых способов улучшить стиль программирования, Пер. с англ. - ДМК, 2006 3. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектноориентированного проектирования, СпБ.: Питер, 2001

4. Слабнов В. Д. Программирование на C++: Учебники и учебные пособия для ВУЗов Познание, 2012

5. Подбельский В.В., Фомин С.С. Курс программирования на языке Си: Учебники Издательство "ДМК Пресс", 2012 https://e.lanbook.com/book/4148

6. Kotlin в действии.Дмитрий Жемеров, Светлана Исакова. 2017

7. П. Дейтел, Х. Дейтел, А. Уолд. Android для разработчиков. 3-е издание(2016)

8. Брайан Харди, Билл Филлипс,Крис Стюарт, Кристин Марсикано. Программирование под Android. 2-е издание (2016)

в) программное обеспечение и Интернет-ресурсы:

1. http://www.omg.org – Object Management Group (Теория объектного подхода, UML)

2. http://cplusplus.com – Информация по языку и стандартной библиотеке C++

3. http://msdm.microsoft.com – Microsoft Developer Network

4. http://www.oracle.com/technetwork/java/index.html - Технология Java

5. Visual Studio 2010 Express – бесплатная среда разработки для C++ под

6. Microsoft Visual Studio 2010 – коммерческая среда разработки для C++ под Windows

7. GNU C++, GNU Make, Eclipse IDE for C++ Developers - свободные средства разработки C++ под Unix/Linux

9. Java SE 6 Development Kit (JDK) – свободно распространяемые средства разработки JavaEclipse IDE for Java Developers, NetBeans IDE – свободные кроссплатформенные среды разработки для Java

8. Материально-техническое обеспечение дисциплины

8.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

8.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

1. Microsoft Windows.
2. Notepad++.
3. Android Studio.
4. Веб-браузер, Chrome.

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, лабораторные работы.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

самоконтроль и самооценка студента;

контроль со стороны преподавателей (текущий и промежуточный).

Критериями оценки результатов самостоятельной работы студента являются:

уровень освоения студентом учебного материала;

умения студента использовать теоретические знания при выполнении практических задач;

сформированность компетенций;

оформление материала в соответствии с требованиями.

10. Методические рекомендации для преподавателя

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

Приложение 1

Структура и содержание дисциплины «Объектно-ориентированное проектирование»
по направлению подготовки
09.03.03 «Прикладная информатика»
Профиль подготовки
Корпоративные информационные системы
(Бакалавр)

n/n	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы Студентов					Формы аттестации	
				Л	П/С	Лаб	СРС	КСР	К.Р.	К.П.	РГР	Реферат	К/р	Э	З
1	Лабораторная работа ЛР-1. Типы данных, операции и условные операторы языка программирования Kotlin	4	1-2			8	8								
2	Лабораторная работа ЛР-2. Циклические алгоритмы. Условия в алгоритмах	4	3-4			8	8								
3	Лабораторная работа ЛР-3. Работа со строками в языке программирования Kotlin.	4	5			4	4								
4	Лабораторная работа ЛР-4. Объектно-ориентированное программирование. Инкапсуляция, полиморфизм и наследование.	4	6-7			8	8								
5	Лабораторная работа ЛР-5. Создание Activity. Жизненный цикл Activity..	4	8-9			8	8								
6	Лабораторная работа ЛР-6. Основы верстки Android приложений.	4	10			4	4								

7	Лабораторная работа ЛР-7. Создание мобильного приложения с использованием ListView и SharedPreferences.	4	11-13			12	12								
8	Лабораторная работа ЛР-8. Современные методы организации сетевого взаимодействия Архитектура REST.	4	14			4	4								
9	Лабораторная работа ЛР-9. Создание мобильного приложения с использованием фрагментов.	4	15			4	4								
10	Лабораторная работа ЛР-10. Создание мобильного приложения с использованием SQLite.	4	16-17			12	12								
	Форма аттестации													**Э**	
	Всего часов по дисциплине					54	90								

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Направление подготовки 09.03.03 Прикладная информатика

профиль подготовки «Корпоративные информационные системы»

Форма обучения: очная

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

Объектно-ориентированное проектирование

Состав:

1. **Показатель уровня сформированности компетенций.**
2. **Перечь оценочных средств.**
3. **Контрольные вопросы.**
4. **Экзаменационное задание.**
5. **Типовой экзаменационный билет.**
6. **Типовые практические задания.**

Москва, 2022 год

1. ПОКАЗАТЕЛЬ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

«Объектно-ориентированное проектирование»					
ФГОС ВО 09.03.03 «Прикладная информатика» профиль подготовки **«Корпоративные информационные системы»**					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **общепрофессиональные компетенции**:					
Компетенции		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства****	**Степени уровней освоения компетенций**
Индекс	**Индекс**				
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение	ИПК-3.1. Знать: методологии разработки программного обеспечения и технологии программирования; методологии и технологии проектирования и использования баз данных; языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования	Лабораторные работы, самостоятельная работа	УО П Зачет	**БАЗОВЫЙ УРОВЕНЬ:** способность выполнять полученное задание, применяя полученные знание и умения на практике, владеть соответствующими индикаторами компетенции при выполнении задания. **ПРОДВИНУТЫЙ УРОВЕНЬ:** способность выполнять полученное задание и решать самостоятельно сформированные задачи, применяя полученные знание и умения на практике. Уверенно владеть соответствующими индикаторами компетенции при выполнении задания,

		программных интерфейсов; методы и средства проектирования баз данных; принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения. ИПК-3.2. Уметь: выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; использовать существующие			комбинировать их между собой и с индикаторами других компетенций для достижения проектных результатов.

		типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ИПК-3.3. Владеть: современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.			
ПК-4	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков	ИПК-4.1. Знать: методы описания алгоритмов, основные абстрактные типы данных и их реализации; проводить работы по интеграции			

	программных продуктов	программных модулей с использованием языков программирования.			

**- Сокращения форм оценочных средств см. в приложении 2 к РП.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Устный опрос / собеседование, (УО)	Средство контроля, организованное как презентация обучающимся результатов выполнения Курсового проекта с демонстрацией наглядных материалов и ответов на вопросы педагогических работников (работника) на тему доклада, теме, проблеме и т.п.	Контрольные вопросы
2	Проект (П)	Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся.	Типовая программа экзамена

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

Примерный перечень вопросов к зачету (экзамену) по всему курсу.

Общая теория ООП.

1. Эволюция методологий программирования.

Парадигмы программирования.

2. Основные принципы объектного подхода. Абстрагирование.

3. Основные принципы объектного подхода. Инкапсуляция.

4. Основные принципы объектного подхода. Модульность.

5. Основные принципы объектного подхода. Иерархия.

6. Основные принципы объектного подхода. Типизация.

7. Основные принципы объектного подхода. Параллелизм. Сохраняемость.

8. Объект с точки зрения ООП. Состояние. Поведение.

9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.

10.Объект с точки зрения ООП. Взаимоотношения между объектами.

11.Классы. Природа классов. Метамодель. Инстанцирование.

12.Классы. Структура класса. Абстрактные классы и интерфейсы.

13.Классы. Отношения между классами. Ассоциация и агрегация.

14.Классы. Иерархии классов. Зависимость.

15. Паттерны проектирования. Общие принципы.

16. Паттерны проектирования. Поиск подходящих объектов.

17. Паттерны проектирования. Определение степени детализации объекта.

18. Паттерны проектирования. Механизмы повторного использования.

19. Паттерны проектирования. Проектирование с учетом будущих изменений.

20. Порождающие паттерны. Абстрактная фабрика (Abstract Factory).

21. Порождающие паттерны. Строитель (Builder).

22. Порождающие паттерны. Фабричный метод (Factory Method).

23. Порождающие паттерны. Прототип (Prototype).

24. Порождающие паттерны. Одиночка (Singleton).

25. Структурные паттерны. Адаптер (Adapter).

26. Структурные паттерны. Мост (Bridge).

27. Структурные паттерны. Компоновщик (Composite).

28. Структурные паттерны. Декоратор (Decorator).

29. Структурные паттерны. Фасад (Facade).

30. Структурные паттерны. Приспособленец (Flyweight).

31. Структурные паттерны. Заместитель (Proxy).

32. Паттерны поведения. Цепочка обязанностей (Chain of

33. Паттерны поведения. Команда (Command).

34. Паттерны поведения. Интерпретатор (Interpreter).

35. Паттерны поведения. Итератор (Iterator).

36. Паттерны поведения. Посредник (Mediator).

37. Паттерны поведения. Хранитель (Memento).

38. Паттерны поведения. Наблюдатель (Observer).

39. Паттерны поведения. Состояние (State).

40. Паттерны поведения. Стратегия (Strategy).

41. Паттерны поведения. Шаблонный метод (Template Method).

42. Паттерны поведения. Посетитель (Visitor).

Средства C++:

1. Модель памяти и структура программы. Классы памяти. Ссылки.

2. Средства абстракции C++. Структура класса. Статические члены.

3. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.

4. Модульность, раздельная компиляция, пространства имен, using директива.

5. Представление иерархических отношений. Наследование.

6. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.

7. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.

8. C++: средства реализации состояния объектов; реализация поведения.

9. Перегрузка операторов.

10. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.

11.Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.

12.Исключения в C++. Обработка исключений.

13.Шаблоны классов и шаблоны функций. Специализация.

14.Основы STL. Структура и назначение. Контейнеры. Алгоритмы

15.Стандартная библиотека, ввод-вывод

Средства Java:

1. Виртуальная машина. Структура программ. Типы переменных в Java. Принципы работы ClassLoader.

2. Средства абстракции Java. Структура класса. Статические члены.

3. Внутренние и вложенные классы. Статический и динамический контекст. Локальные и анонимные классы. Перечисления

4. Средства инкапсуляции Java. Поддержка модульности. Пакеты.

5. Представление иерархических отношений. Наследование. Интерфейсы и абстрактные классы.

6. Родовые компоненты (Generics)

7. Агрегация и зависимость от времени жизни. Реализация отношений клиентсервер. Стандартные контейнеры.

9. Типизация. Правила преобразования типов. instanceof и средства поддержки параллелизма. Активные и пассивные объекты. Класс Object.

10.Использование Thread и Runnable. Пул потоков, назначение и принципы реализации.

11.Исключения. Обработка исключительных ситуаций.

12.Сохраняемость. Serializable и Externalizable. Программирование распределенных приложений.

13.Модель безопасности Java. Policy, Permissions, AccessController.

14.Графическая подсистема. Основы AWT, Applet, Swing components. Событийная модель.

15.Средства поддержки Java машины. System, Runtime, сборка мусора.

4. ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ

Экзаменационное задание выполняется студентом индивидуально, по итогам изучения дисциплины или ее части. При этом достижение порогового результата работы над экзаменационным заданием соответствует описанному в п. 3 данного документа этапу освоения соответствующих компетенций на базовом или продвинутом уровне.

Базовый уровень: способность выполнять полученное задание, применяя полученные знание и умения на практике, владеть соответствующими индикаторами компетенции при выполнении задания.

Продвинутый уровень: способность выполнять полученное задание и решать самостоятельно сформированные задачи, применяя полученные знание и умения на практике. Уверенно владеть соответствующими индикаторами компетенции при выполнении задания, комбинировать их между собой и с индикаторами других компетенций для достижения проектных результатов.

Форма экзаменационного задания выбирается преподавателем и утверждается на заседании кафедры. Экзамен может проходить в следующих формах и с использованием следующих оценочных средств.

Форма	Представление оценочного средства в ФОС
Устная.	Банк контрольных вопросов, соответствующих отдельным темам дисциплины (см. п. 4 настоящего документа). Вопросы формируют экзаменационный билет (см. ниже), состоящий из теоретических вопросов и практических заданий (типовые практические задания представлены ниже). Билеты, включая вопросы и практические задания, формируются преподавателем и утверждаются на заседании кафедры. В них могут быть включены дополнительные контрольные вопросы и задания, не требующие у студентов наличия не формируемых данной дисциплиной компетенций или более высоких этапов сформированности формируемых. Для ответа на каждый вопрос и для решения любого практического задания студент должен находится на требуемом для данной дисциплине уровне сформированности всех соответствующих ей компетенций: каждый вопрос и задание проверяет уровень сформированности всех соответствующих данной дисциплине компетенций.
Письменная.	Оценочное средство полностью соответствует оценочным средствам устной формы задания.

Практико- ориентированная (формат WorldSkills).	Типовое задание практико-ориентированного экзамена. Задание практико-ориентированного формируется преподавателем на основе типового и Методических рекомендаций по разработке задания ПОЭ, утверждаются на заседании кафедры. Задание ПОЭ проверяет уровень сформированности всех соответствующих дисциплине компетенций.

5. ТИПОВОЙ ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1
по дисциплине
«Объектно-ориентированное проектирование»
направление подготовки 09.03.03 Прикладная информатика

ВОПРОСЫ:

1. В чем отличие статического и динамического контента?
2. Сколько аргументов может быть передано функции?
3. Какие параметру присутствуют у функции microtime() и для чего они используются?
4. Практическое задание. Разработать программу-обработчик полученных оценок за экзамен. На входе через GET-параметры последовательно передается информация об оценке и ФИО студента. После каждой обработки должна выводится столбчатая диаграмма с распределением оценок. Под каждым столбцом выводятся соответствующие ФИО

Утверждено: _____ / _____ / «__» _____ 20__г.

6. ТИПОВЫЕ ПРАКТИЧЕСКИЕ ЗАДАНИЯ

и Разработать программу-обработчик полученных оценок за экзамен. На входе через GET-параметры последовательно передается информация об оценке ФИО студента. После каждой обработки должна выводится столбчатая диаграмма с распределением оценок. Под каждым столбцом выводятся соответствующие ФИО.

2. Разработать программу, формирующую для указанного URL список располагающихся на соответствующей веб-странице уникальных ссылок.

3. Разработать программу, строящую гистограмму количества слов в тексте без учета регистра символов.

4. Разработать программу, реализующую алгоритм чет-нечет.

5. Разработать программу, загружающую файл со списком формата "e-mail: сообщение" и рассылающую адресатам соответствующие сообщения.

Типовое задание в практико-ориентированном формате

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ
по дисциплине
«Объектно-ориентированное проектирование»
направление подготовки 09.03.03 Прикладная информатика

ВРЕМЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

6 астрономических часов без учета перерыва на обед.

ЗАДАНИЕ ЭКЗАМЕНА

Компания Marvel, впечатленная достигнутыми результатами в разработки мобильных приложений, поручает Вам создать мобильное приложение, состоящие из следующих экранов:

Экран ввода пароля для входа в
приложение; Экран со списком комиксов;
Экран с детальной информацией о комиксе;
Экран со списком персонажей комиксов;
Экран с детальной информацией о персонаже комикса;
Экран со списком избранных персонажей и комиксов;
Экран с информацией о приложении.

1. Содержание экрана ввода пароля в приложение:
   1.1. Логотип
   1.2. Поле ввода пароля(пароль - admin);
   1.3. Кнопка входа.
2. Содержание элемента списка персонажей / комиксов:
   2.1. Изображение персонажа / комикса;
   2.2. Наименование персонажа / комикса;
   2.3. Количество комиксов с персонажем / цена комикса;
   2.4. Кнопка для добавления в избранное.
3. Содержание экрана с детальной информацией персонажа / комикса: 3.1. Изображение персонажа / комикса;
   3.2. Наименование персонажа / комикса;
   3.3. Полное описание персонажа / комикса;
   3.4. Кнопка для добавления в избранное;
   3.5. Количество комиксов с персонажем / цена комикса.
4. Содержание экрана со списком избранных персонажей и комиксов: 4.1. Список избранных персонажей и комиксов.

5. Содержание экрана с информацией о приложении: 5.1. Логотип;
   5.2. Информация о приложении;
   5.3. Дата создания приложения;
   5.4. Информация о Вас.

Переходы между экранами должны быть реализованы через нижнюю панель навигации (BottomNavigationView), позволяющая переключаться между экранами приложения в одно касание.

В рамках поставленной задачи необходимо реализовать на некоторых экранах следующие функции.

1. На экране ввода пароля для входа в приложение:
   1.1. сохранение входа в приложение;
2. На экране с детальной информацией персонажа / комикса:
   2.1. Добавление в избранное;
   2.2. Удаление из избранного.
3. На экране со списком персонажей / комиксов:
   3.1. открытие экрана с детальной информацией об элементе;
   3.2. сортировка списка персонажей по алфавиту (по возрастанию и убыванию);
   3.3. сортировка списка комиксов по цене (по возрастанию и убыванию).

Приложение должно уметь кешировать данные, то есть работать в offline режиме.

ВХОДНЫЕ ДАННЫЕ

В качестве исходного материала для работы необходимо использовать:

Предоставленные медиа файлы(изображения и т д).;
Документация Marvel API;
Ключи доступа к Marvel API.

ВЫХОДНЫЕ ДАННЫЕ

В результате работы в папке "*Рабочий стол/ФИО - Номер группы* " должны быть предоставлены следующие материалы:

Архив проекта Android Studio;

Apk-файл созданного приложения;

УСЛОВИЯ ВЫПОЛНЕНИЯ РАБОТЫ

Для выполнения задания может использоваться следующее программное обеспечение:

Android Studio;

Notepad++;

Adobe Photoshop;

Браузер.

Не разрешается запускать и использовать другие программы. Не допускается использование *flash*-накопителей, материалов на серверах. Разрешается использование лекций учебного курса и печатных учебников по Java, Android Studio, Разработки мобильных приложений под ОС Android.

КРИТЕРИИ ОЦЕНКИ ЗАДАНИЯ

При проверке работы будет последовательно выполнены описанные в таблице действия. При соответствии выполняемого действия ожидаемому результату подтверждается указанная в правой колонке оценка. В случае ошибочного результата или отсутствия реализации проверяемой функции, вследствие чего выполнить требуемые действия невозможно, итоговая оценка экзамена будет взята как предыдущая подтвержденная оценка. В этом случае дальнейшая проверка работы осуществляться не будет.

№	Выполняемые действия	Проверяемая функция	Ожидаемый результат	Оценка
1.	Просмотр строковых ресурсов проекта(/res/values/strings.xml)	Все строки вынесены в ресурсы	Все строки, используемые в приложение находятся в ресурсах(/res/values/strings.xml)	неудовлетворительно
2.	На экране ввода пароля для входа в приложение нажатие на кнопку входа(с соответствующей надписью или графическим обозначением)	Вход в приложение	При вводе верного пароля происходит вход в приложение, при вводе неверного пароля отображается уведомление с соответствующим сообщением.	
3.	Повторный вход в приложение	Повторный вход в приложение	При повторном входе в приложение пароль не запрашивается	
4.	Нажатие на нижнюю панель навигации	Открытие экрана	Наличие всех пунктов меню для перехода на другой экран	
5.	Открытие экрана о приложении	Открытие экрана	Наличие экрана о приложении	удовлетворительно

6.	Отображение списка персонажей комиксов	Корректное отображение элемента списка	Список реализован при помощи виджета ListView либо RecyclerView. Внешний вид элемента списка не меняется при разных размерах экрана устройства.		
7.	Нажатие на элемент списка персонажей комиксов	Переход на экран детального отображения	Реализован интерфейс OnItemClickListener для перехода на экран детального отображения		
8.	Отображение списка комиксов	Корректное отображение элемента списка	Список реализован при помощи виджета ListView либо RecyclerView. Внешний вид элемента списка не меняется при разных размерах экрана устройства.	0 баллов	
9.	Нажатие на элемент списка комиксов	Переход на экран детального отображения	Реализован интерфейс OnItemClickListener для перехода на экран детального отображения		
10.	Нажатие на кнопку избранное на элементе списка	Добавление / удаление элемента	Реализована логика добавления в избранное и удаления из избранного		
11.	Открытие экрана избранное	Открытие экрана	Отображение списка с избранными элементами		

12.	Активаций элемента управления (кнопка или элемент списка) "Сортировка по алфавиту по возрастанию" (с соответствующей надписью или графическим обозначением)	Сортировка списка персонажей по алфавиту (по возрастани ю)	Строки списка сортируются по алфавиту в порядке возрастания. Содержимое каждой строки не изменяется – меняется лишь порядок строк		
13.	Активаций элемента управления (кнопка или элемент списка) "Сортировка по алфавиту по убыванию " (с соответствующей надписью или графическим обозначением)	Сортировка списка персонажей по алфавиту (по убыванию)	Строки списка сортируются по алфавиту в порядке возрастания. Содержимое каждой строки не изменяется – меняется лишь порядок строк		
14.	Активаций элемента управления (кнопка или элемент списка) "Сортировка по цене по возрастанию" (с соответствующей надписью или графическим обозначением)	Сортировка списка комиксов по цене (по возрастани ю)	Строки списка сортируются по алфавиту в порядке возрастания. Содержимое каждой строки не изменяется – меняется лишь порядок строк		
15.	Активаций элемента управления (кнопка или элемент списка) "Сортировка по цене по убыванию " (с соответствующей надписью или графическим обозначением)	Сортировка списка комиксов по цене (по убыванию)	Строки списка сортируются по алфавиту в порядке возрастания. Содержимое каждой строки не изменяется – меняется лишь порядок строк		
16.	Открытие приложения без доступа к Интернету	Кеширование информации	Отображение всех персонажей, комиксов без доступа в интернет, кроме первого запуска приложения		

17.	Просмотр архитектуры приложения	Реализация одного из паттернов проектирования MVP/ MVVM	Реализован один из паттернов проектирования MVP/ MVVM	

Набранный уровень достижения, и соответствующая им оценка имеет рекомендательный характер – экзаменатор имеет право скорректировать оценку в ту или иную сторону.