

МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Директор института прикладной
математики и компьютерных наук
А.В. Замятин
« 14 » _____ 2023 г.



Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине
(Оценочные средства по дисциплине)

Разработка приложений для мобильных платформ

по направлению подготовки

09.03.03 Прикладная информатика

Направленность (профиль) подготовки :

Разработка программного обеспечения в цифровой экономике

Томск–2023

ОС составил(и):
ассистент кафедры программной инженерии



Л.С. Иванова

Рецензент:
д-р физ.-мат. наук, профессор,
профессор кафедры программной инженерии



О.А. Змеев

Оценочные средства одобрены на заседании учебно-методической комиссии
института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от 08.06.2023г. №2

Председатель УМК ИПМКН,
д-р техн. наук, профессор



С.П. Сущенко

Оценочные средства (ОС) являются элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ОС разрабатывается в соответствии с рабочей программой (РП) дисциплины и включает в себя набор оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине.

1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Компетенция	Индикатор компетенции	Код и наименование результатов обучения (планируемые результаты обучения, характеризующие этапы формирования компетенций)	Критерии оценивания результатов обучения	
			Зачтено	Не зачтено
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ИОПК-7.1 Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем ИОПК-7.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий	ОР-7.1.1. Умеет реализовывать различные алгоритмы в приложениях для мобильных платформ. ОР-7.2.1. Умеет использовать фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий	Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем, использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий Умеет реализовывать различные алгоритмы в приложениях для мобильных платформ. Умеет использовать фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий	Не использует методы построения и анализа алгоритмов при проектировании и разработке программных систем, не использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий Не умеет реализовывать различные алгоритмы в приложениях для мобильных платформ Не умеет использовать фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий

	ИОПК-7.3 Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности	ОР-7.3.1. Знает синтаксис языков, применяемых для разработки мобильных приложений.	Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности Знает синтаксисом языка, используемого для разработки мобильных приложений.	Не разрабатывает алгоритмы и программы при решении задач профессиональной деятельности Не знает синтаксисом языка, используемого для разработки мобильных приложений.
ПК-1 – Способен осуществлять программирование, тестирование и опытную эксплуатацию ИС с использованием технологических и функциональных стандартов, современных моделей и методов оценки качества и надежности программных средств	ИПК-1.1 Определяет, согласовывает и утверждает требования заказчика к ИС	ОР-1.1.1. Умеет создавать программы для мобильных платформ. ОР-1.2.1. Умеет проектировать программное обеспечение	Определяет, согласовывает и утверждает требования заказчика к ИС, проектирует программное обеспечение Умеет создавать программы для мобильных платформ. Умеет проектировать программное обеспечение	Не определяет, согласовывает и не утверждает требования заказчика к ИС, не проектирует программное обеспечение Умеет создавать программы для мобильных платформ. Не умеет создавать программы для мобильных платформ. Не умеет проектировать программное обеспечение
	ИПК-1.3 Кодирует на языках программирования и проводит модульное тестирование ИС	ОР-1.3.1. Умеет выбирать инструментарий для реализации мобильного приложения, наиболее подходящий для конкретной задачи.	Кодирует на языках программирования и проводит модульное тестирование ИС Умеет выбирать инструментарий для реализации мобильного приложения, наиболее подходящий для конкретной задачи.	Не кодирует на языках программирования и проводит модульное тестирование ИС Не умеет выбирать инструментарий для реализации мобильного приложения, наиболее подходящий для конкретной задачи.

2. Этапы формирования компетенций и виды оценочных средств

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Введение в разработку для мобильных платформ. Синтаксис языка.	ОР-7.1.1, ОР-7.2.1, ОР-7.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
2.	Архитектурные особенности приложений для мобильных платформ.	ОР-7.1.1, ОР-7.2.1, ОР-7.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
3.	Базовые элементы пользовательского интерфейса.	ОР-7.1.1, ОР-7.2.1, ОР-7.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
4.	Списочные элементы пользовательского интерфейса. Анимация.	ОР-7.1.1, ОР-7.2.1, ОР-7.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
5.	Хранение данных.	ОР-7.1.1, ОР-7.2.1, ОР-7.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
6.	Особенности разработки приложений для современных версий мобильных операционных систем.	ОР-7.1.1, ОР-7.2.1, ОР-7.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы

3. Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

3.1. Типовые задания для проведения текущего контроля успеваемости по дисциплине.

Фонд оценочных средств, для проведения текущего контроля успеваемости включает в себя:

- устные опросы;
- задания для самостоятельной работы;
- описание процедуры оценивания.

Примеры вопросов и заданий для устного опроса:

Тема №1.

1. Назовите основные слои абстракции операционной системы iOS.
2. Какие основные компоненты доступны на верхнем уровне абстракции Android?
3. К какому классу языков программирования относится Swift?
4. Какие классы коллекций доступны в языке Swift? Каковы их особенности?
5. Какие языковые конструкции и типы данных доступны в языке Kotlin?
6. Каковы особенности работы с необязательными типами на языке Kotlin?
7. Какие основные варианты представления данных используются в Swift? В чем их сходство и отличие?
8. Что такое делегирование? Какие проблемы можно решить с помощью данного шаблона?
9. Какие специфические для функциональных языков конструкции доступны в языке Kotlin?
10. Что такое циклическая зависимость? Какие способы устранения данной ситуации доступны в языке Swift?

Тема №2

1. В чем особенности архитектурного подхода Cocoa MVC?
2. Каковы основные этапы жизненного цикла экземпляров класса UIViewController?
3. Каковы основные этапы жизненного цикла приложения для операционной системы Android?
4. Каково назначение классов UIApplication и UIApplicationDelegate?
5. Охарактеризуйте один из шаблонов Cocoa.

Тема №3

1. Что происходит с view на экране при повороте устройства?
2. Какие структуры данных используются для определения положения объекта на экране?
3. Какую информацию содержат свойства bounds и frame (класс UIView)?
4. Какие основные методы нужно реализовать при создании собственного подкласса UIView?
5. В чем различия между дискретными и непрерывными жестами? Приведите примеры жестов каждой группы.
6. Какие способы смены экрана доступны в Interface Builder?
7. Какие инструменты доступны разработчику для работы в autolayout?

Тема №4

1. В чем отличие между классами UITableView и UITableViewController?
2. Какие протоколы необходимо реализовать для корректной работы UITableView?
3. Для чего нужен компонент UICollectionView?
4. Какие виды анимации доступны при разработке для Android?
5. Каким образом происходит настройка положения view в UIScrollView?

Тема №5

1. В отличия между локализацией и интернационализацией? Каким образом осуществляется интернационализация приложений для iOS? А для Android?
2. Какие датчики доступны для работы на устройствах под управлением iOS? А Android?
3. Какие источники данных доступны в Android для определения местоположения устройства? В чем их отличия?
4. Какую информацию содержат данные, получаемые от акселерометра?
5. Опишите по шагам процесс получения фотографии из альбома в приложении для iOS.

Тема №6

1. Каковы основные отличия интерфейса приложений для мобильных устройств от приложений для ПК?
2. В чем различия между режимами мультитасчности Slide Over и Split View?
3. Что такое 3D-touch? Для чего данная технология применяется в iOS?
4. Каковы особенности операционной системы WatchOS? Для каких устройств она предназначена?
5. Что такое TVML? Для чего он применяется?

Описание лабораторных работ.

Язык программирования – Kotlin.

Написать приложение, отображающее список элементов на одной Activity (ListActivity) с возможностью просмотреть подробную информацию об элементе списка на другой Activity (DetailsActivity). В качестве таких элементов может выступать кто угодно – музыканты, актёры, политики или что угодно – организации, вещи и т. д. Каждому элементу должны соответствовать, как минимум, иконка, название, описание рейтинг (float в интервале [0; 5]). В ListActivity достаточно отображать иконку, название и рейтинг (при наличии). В DetailsActivity должна быть возможность изменить рейтинг. Начальный список элементов должен храниться локально в каком-то хранилище (Preferences, файл или база данных), а затем обновляться с сервера. При этом на сервере не будет рейтинга элемента, рейтинг исключительно локальный. Эти изменения должны быть отображены в ListActivity динамически. Данные можно разместить, например, на Google Disk, получив постоянную ссылку на него. Обращение к нему должно происходить посредством Retrofit или OkHttp. Список элементов в ListActivity должен отображаться с помощью RecyclerView. Асинхронные вызовы выполнять с помощью kotlin coroutines. Для каждой Activity нужно реализовывать свой ViewModel.

Последовательность задач при выполнении лабораторных работ

1. Создать ListActivity. Набор элементов генерируется. Сразу должна присутствовать ViewModel с LiveData внутри.
 - a. Добавить сверху экрана надпись с количеством элементов (использовать plurals).
2. Набор элементов для ListActivity получается из локального хранилища.
3. Создать DetailsActivity.
4. Добавить Dependency Injection, реализованный с помощью Dagger.
5. Реализовать получение элементов с сервера, иконки при этом заданы в виде ссылок, которые пока не используются. Если с сервера подгружен элемент, который не хранится локально, для него используется иконка по умолчанию.
6. Загружать иконки по ссылкам динамически с помощью Glide.
 - a. Добавить предзагрузку иконок путём интеграции Glide и RecyclerView.
7. Авторизация. Добавить 2 активности – для регистрации SignUpActivity и для ввода пароля при входе LoginActivity. Проверять пароль автоматически при вводе пользователем значения с задержкой в 150 мс. Пароль на этом этапе можно просто сохранить в Preferences. Если пользователь забыл пароль, должна быть возможность стереть все данные и перейти к SignUpActivity.
 - a. Добавить авторизацию с помощью TouchId.
 - b. Если пользователь вышел из приложения более, чем на 5 минут, показывать при возвращении LoginActivity.
8. Добавить новый тип сборки (build type) «staging». Использовать различные ссылки на файл с элементами для различных типов сборки.

Ввиду практической направленности курса, зачет как таковой не проводится. Вместо этого аттестация осуществляется по результатам выполнения лабораторных работ. Оценка «отлично» («зачтено») ставится при выполнении 7-8 заданий из приведённого выше списка, оценка «хорошо» («зачтено») ставится при выполнении 5-6 заданий и, наконец, «удовлетворительно» («зачтено») – при выполнении 4 заданий.

Задания для самостоятельной работы:

Задание №1. «Сортировка массивов».

Цель работы – реализовать один из алгоритмов сортировки массивов (Шелла, пирамидальная, быстрая).

Ввод данных должен осуществляться через интерфейс пользователя, значения необходимо вводить через запятую или пробел. Сортировка не должна осуществляться в главном потоке приложения. Вывод результата в произвольной форме.

Сдача выполненного задания производится в форме защиты работающего приложения и его исходного кода.

Задание №2 «Курс валют».

Цель работы – спроектировать и реализовать приложение, запрашивающее данные о текущем курсе валют в сети Интернет.

Информация от источника данных должна быть представлена в формате JSON. На экране смартфона необходимо отобразить информацию о любой паре валют. Запрос данных не должен осуществляться в главном потоке приложения. Вывод результата в произвольной форме.

Сдача выполненного задания производится в форме защиты работающего приложения и его исходного кода.

Задание №3 «Список дел».

Цель работы – спроектировать и реализовать приложение для отображения списка текущих дел.

Список дел должен отображаться в виде таблицы, высота ячейки таблицы должна зависеть от содержимого. Необходимо отображать следующую информацию: название, статус дела, текстовое описание, дату создания, важность дела. Добавление дела в таблицу должно осуществляться в отдельном окне.

Сдача выполненного задания производится в форме защиты работающего приложения и его исходного кода.

Задание №4 «Список дел с дополнительными возможностями».

Цель работы – на основе результатов выполнения задания №3 создать приложение с дополнительными возможностями: добавление фотографий к списку дел, валидация вводимой информации, наличие англоязычной версии приложения.

Дела из списка должны подлежать редактированию. Необходимо так же реализовать цветовую дифференциацию дел в списке по важности. Добавление фотографий должно осуществляться как из альбома устройства, так и с камеры.

Сдача выполненного задания производится в форме защиты работающего приложения и его исходного кода.

4. Методические материалы, определяющие процедуры оценивания образовательных результатов обучения

4.1. Методические материалы для оценки текущего контроля успеваемости по дисциплине.

Текущий контроль по лабораторным работам осуществляется в виде проверки выполнения заданий лабораторной работы. Текущий контроль успеваемости по теоретическому материалу осуществляется в виде устных опросов.

Оценка текущего контроля проводится на основе оценки компетенций, соответствующих текущему разделу дисциплины, согласно таблице раздела 1.

4.2. Методические материалы для проведения промежуточной аттестации по дисциплине.

Зачет в восьмом семестре проводится в форме сдачи практической работы.

Итоговая оценка по предмету вычисляется как среднее арифметическое значение оценок за практические работы при условии, что все оценки положительные. В случае получения за одну из работ неудовлетворительной оценки выставляется оценка «неудовлетворительно».