



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.01.09 Разработка мобильных приложений

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Компьютерное моделирование и анализ данных

Квалификация (степень): бакалавр

Форма обучения: очная

Институт: математики, естествознания и техники

Кафедра: математического моделирования, компьютерных технологий и информационной безопасности

	очная форма	очно-заочная форма	заочная форма
Курс	2,3		
Семестр/триместр	4, 5, 6		
Лекции	54		
Лабораторные занятия	54		
Практические (семинарские) занятия	54		
в т. ч. практическая подготовка	12		
Форма(ы) промежуточной аттестации	Зачет Зачет с оценкой Экзамен – 0,3		
Контроль	18		
Иные формы работы	-		
Самостоятельная работа	215,4		

Всего часов: **396**

Трудоемкость: **11** зачетные единицы.

Разработчик(и) рабочей программы:

Ассистент кафедры ММКТиИБ

Черепков А.Ю

I. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

Цель изучения дисциплины: формирование навыков проектирования приложений и разработки пользовательского интерфейса в контексте разработки мобильных приложений, получения теоретической и практической подготовки обучающихся в области разработки на мобильных платформах.

Задачи изучения дисциплины:

- дать представления о принципах разработки мобильных приложений;
- рассмотреть основные виды мобильных платформ;
- научить применять полученные знания в профессиональной деятельности.

Место дисциплины в структуре ОПОП: реализуется в рамках части, формируемой участниками образовательных отношений, блока Б1. Дисциплины (модули).

Планируемые результаты обучения по дисциплине:

Код компетенции	Индикаторы компетенции	Планируемые результаты обучения по дисциплине
ПКС-1	Знать: <ul style="list-style-type: none">– возможности существующей программно-технической архитектуры;– методологию разработки программного обеспечения и технологию программирования;– методы и средства проектирования программного обеспечения;– типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения.	Знает: <ul style="list-style-type: none">– методологию разработки приложений для мобильных устройств.
	Уметь: <ul style="list-style-type: none">– проводить оценку и обоснование рекомендуемых решений;– вырабатывать варианты реализации программного обеспечения;– применять методы и средства проектирования программного обеспечения, структур данных, программных интерфейсов.	Умеет: <ul style="list-style-type: none">– применять методы и средства проектирования мобильных приложений.

Владеть: <ul style="list-style-type: none"> – анализом возможностей реализации требований к программному обеспечению; – навыками распределения заданий между программистами в соответствии с техническими спецификациями; – методами проектирования структур данных; – методами проектирования программных интерфейсов; – навыками осуществления обучения и наставничества. 	Владеет: <ul style="list-style-type: none"> – навыками разработки приложений для мобильных платформ.
---	--

II. СОДЕРЖАНИЕ И ОБЪЕМ ДИСЦИПЛИНЫ

с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу

Очная форма обучения

№ п/п	Наименование разделов и тем	Всего	Аудиторные занятия			Сам. раб.
			ЛК	ПЗ	ЛБ	
4 семестр						
	Раздел 1. «Мобильные платформы, обзор возможностей»	24	4	4	4	12
1	Тема 1. «Сравнительный обзор мобильных операционных систем»	12	2	2	2	6
2	Тема 2. «Особенности разработки	12	2	2	2	6
	Раздел 2. «Основы Kotlin»	84	14	14	14	42
3	Тема 1. «Структура программы»	12	2	2	2	6
4	Тема 2. «Основные конструкции»	12	2	2	2	6
5	Тема 3. «Массивы»	12	2	2	2	6
6	Тема 4. «Функционального программирования	12	2	2	2	6
7	Тема 5. «Классы и объекты»	12	2	2	2	6
8	Тема 6. «Наследование»	12	2	2	2	6
9	Тема 7. «Обобщенные типы»	12	2	2	2	6
	Контроль					
	Форма отчетности		зачет			
	Итого за 5 семестр	108	18	18	18	54
5 Семестр						
10	Раздел 2. «Основы Kotlin»	44	4	4	4	32
11	Тема 8. «Коллекции»	22	2	2	2	16
12	Тема 9. «Асинхронные операции»	22	2	2	2	16
	Раздел 3. «Разработка мобильных приложений под Android»	90,7	14	14	14	48,7
13	Тема 1. «Создание приложения и запуск на устройстве»	34	6	6	6	16

14	Тема 2. «Макеты экрана приложения и ресурсы»	28	6	4	4	16
15	Тема 3. «Основные элементы управления»	28,7	2	4	4	16,7
16	Контроль	9				
	Форма отчетности	0,3	Экзамен			
	Итого за 6 семестр	144	18	18	18	80,7
6 семестр						
	Раздел 3. «Разработка мобильных приложений под Android»	144	18	18	18	80,7
	Тема 1. «Основные элементы управления»	16	2	2	2	10
17.	Тема 2. «Обработка событий»	14	2	2	2	8
18.	Тема 3. «Жизненный цикл активности»	14	2	2	2	8
19.	Тема 4. «Привязка данных»	14	2	2	2	8
20.	Тема 5. «Навигация внутри приложения»	14	2	2	2	8
21.	Тема 6. «Анимированные переходы»	14	2	2	2	8
22.	Тема 7. «Передача данных между активностями»	16	2	2	2	10
23.	Тема 8. «Работа с базами данных»	16	2	2	2	10
24.	Тема 9. «Работа с сетью»	16,7	2	2	2	10,7
	Форма отчетности	0,3	Экзамен			
	Контроль	9				
	Итого за 7 семестр	144	18	18	18	80,7
	в т.ч. практическая подготовка	12				
	ИТОГО:	396	54	54	54	215,4

Очно-заочная форма обучения (не реализуется)

Заочная форма обучения (не реализуется)

III. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Текущая аттестация проводится в форме теста.

Типовой вариант тестовый заданий

Что из этого в настоящее время не поддерживается в Kotlin?

1. JVM
2. JavaScript
3. LLVM
4. **.NET CLR**

Какое выражение Kotlin эквивалентно данному из C?

int x = a ? b : c

1. val x = a ?: b, c
2. val x = if (a) b : c
3. val x = a ? b : c
4. **val x = if (a) b else c**

Что применимо для следующего объявления класса?

1. Он package-private
2. Он может быть расширен другими классами

3. Он public

4. У него приватное свойство "name"

Есть ли у Kotlin примитивные типы данных, такие как int, long, float?

1. Нет, Kotlin не имеет и не использует примитивные типы данных.
2. **Нет, не на уровне языка. Но компилятор Kotlin использует примитивы JVM для лучшей производительности.**
3. Да, но Kotlin всегда конвертирует их в не примитивные аналоги.
4. Да, Kotlin в этом отношении похож на C.

Что такое to в приведенном ниже примере: val test = 33 to 42

1. **Инфиксная функция, создающая пару (33, 42)**
2. Ключевое слово Kotlin для создания пары (33, 42)
3. Ключевое слово для создания диапазона от 33 до 42
4. Опечатка

Какое из объявлений функций является валидным?

1. int sum(int a, int b)
2. int sum(a: Int, b: Int)
3. function sum(a: Int, b: Int): Int
4. **fun sum(a: Int, b: Int): Int**

В чем ключевое отличие Iterable<T> и Sequence<T> в Kotlin?

1. Iterable<T> работает только с immutable коллекциями, Sequence<T> применим к mutable
2. Нет никакой разницы, т. к. Sequence<T> аналог Iterable<T>
3. **Последовательности обрабатываются лениво, итераторы жадно**
4. Последовательности обрабатываются по очереди, итераторы параллельно (многопоточно)

Чего не предлагает dataclass?

1. Авто-генерируемый метод toString()
2. Метод сору(...), для создания копии экземпляров.
3. **Автоматическое преобразование из/в JSON**
4. Авто-генерируемые методы hashCode() и equals()

Что выведет следующий код?

```
val listA = mutableListOf(1, 2, 3)
val listB = listA.add(4)
print(listB)
```

1. [1, 2, 3, 4]
2. **True**
3. Ничего, тут ошибка компиляции
4. Unit

В чем разница между a и b?

```
var a: String? = "KotlinQuiz"
var b: String = "KotlinQuiz"
```

1. a является volatile, как в Java
2. b является final и не может быть изменено
3. a является final и не может быть изменено
4. **b никогда не сможет стать null**

Как в Kotlin правильно объявить переменную целочисленного типа?

1. var i : int = 42
2. let i = 42
3. int i = 42
4. **var i : Int = 42**

Какой тип у arr?

val arr = arrayOf(1, 2, 3)

1. **Array<Int>**
2. Int[]
3. int[]
4. IntArray

Что из приведенного в Kotlin описывает статический метод?

1. class Foo {
 @static fun bar() : String = "Kotlin"
}
2. **class Foo {
 companion object {
 fun bar() : String = "Kotlin"
 }
}**
3. class Foo {
 static fun bar() : String = "Kotlin"
}

Для чего нужен оператор !! ?

1. Он возвращает левый операнд, если он не равен null, иначе возвращает правый операнд
2. Это оператор модуля, аналог % в С.
3. Он сравнивает два значения на тождественность
4. **Он преобразует любое значение в ненулевой тип и выбрасывает исключение, если значение равно null**

Укажите правильный синтаксис для преобразования строки "42" в long

1. val l: Long = <Long>"42"
2. **val l: Long = "42".toLong()**
3. val l: Long = (Long)"42"
4. val l: Long = Long.parseLong("42")

Что такое корутины (coroutines)?

1. Функции, которые принимают другие функции в качестве аргументов или возвращают их.
2. **Методы, обеспечивающие асинхронный код без блокировки потока.**
3. Термин Kotlin, используемый в методах класса.
4. Автоматически сгенерированные методы hashCode() и equals() в data classes.

Что делает этот код?

foo()

1. Создает двумерный массив
2. Не скомпилируется
3. Вызывает асинхронно foo
4. **Вызывает функцию, которая вернется после вызова foo**

В чем разница между val и var в Kotlin?

1. **Переменные, объявленные с помощью val, являются final, а переменные var – нет.**
2. Переменные, объявленные с помощью val, имеют доступ только к const членам.
3. Переменные, объявленные с помощью var, являются final, а переменные val – нет.
4. var ограничен видимостью ближайшего функционального блока, а у val видимость заканчивается на ({ }).

Что выведет следующий код?

```
val list : List<Int> = listOf(1, 2, 3)
list.add(4)
```

```
print(list)
```

1. Он не компилируется, так как List не имеет метода add

2. [1, 2, 3, 4]

3. [5, 6, 7]

4. Он не компилируется, из-за listOf

Что выведет этот код?

```
val a: String? = null
```

```
val b: String = "Hello World"
```

```
println(a==b)
```

1. Ничего, т. к. вылетит исключение NullPointerException

2. Не скомпилируется

3. false

4. true

Промежуточная аттестация обучающихся осуществляется в форме зачета, зачета с оценкой и экзамена с использованием следующих оценочных материалов: вопросы к зачету, вопросы к зачету с оценкой, вопросы к экзамену.

Вопросы к зачету (4 семестр, очная форма обучения)

1. Основные характеристики мобильных платформ
2. Структура программы
3. Переменные и типы данных
4. Условные конструкции
5. Циклы
6. Последовательности
7. Массивы
8. Функции и их аргументы
9. Переменное число аргументов
10. Перегрузка функций
11. Анонимные функции
12. Лямбда-выражения

Вопросы к зачету с оценкой (5 семестр, очная форма обучения)

1. Классы и объекты
2. Пакеты
3. Наследование
4. Модификаторы видимости
5. Переопределение методов
6. Абстрактные классы
7. Интерфейсы
8. Перечисления
9. Анонимные классы
10. Обобщенные классы
11. Обработка исключений
12. Списки
13. Корутины
14. Асинхронные потоки

Вопросы к экзамену (6 семестр, очная форма обучения)

1. Типы приложений
2. Виды макетов экрана
3. Поля ввода текста
4. Переключатели
5. Всплывающие окна
6. События
7. Жизненный цикл активности
8. Запуск активности
9. Передача данных между активностями
10. Подключение к базе данных
11. Получение данных
12. Манипулирование данными
13. Модели данных
14. Отображение веб-страницы
15. Настройка соединения
16. Отправка запросов на сервер

IV. ПЕРЕЧЕНЬ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Основная литература

1. Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений: учебное пособие для вузов / В. В. Соколова. — Москва: Издательство Юрайт, 2024. — 160 с. — (Высшее образование). — ISBN 978-5-534-16302-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/537272> (дата обращения: 20.04.2024).
2. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва: Издательство Юрайт, 2024. — 160 с. — (Профессиональное образование). — ISBN 978-5-534-16868-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/542342> (дата обращения: 20.04.2024).

Дополнительная литература

1. Kotlin для Android : от теории к практике : учебно-методическое пособие : [16+] / сост. А. В. Конаков. — Братск : Братский государственный университет (БрГУ), 2023. — 54 с.: ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=701396> (дата обращения: 20.04.2024). — Библиогр. в кн. — Текст : электронный.
2. Введение в разработку приложений для ОС Android / Ю. В. Березовская, О. А. Юфрякова, В. Г. Вологодина и др. — 2-е изд., испр. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016. — 434 с.: ил. — URL: <https://biblioclub.ru/index.php?page=book&id=428937> (дата обращения: 20.04.2024).

V. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО- ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ

ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№ п/п	Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
1.	http://edu.ru/	Российское образование: Федеральный портал. Включает ссылки на порталы и сайты образовательных учреждений; государственные образовательные стандарты; нормативные документы; каталог экскурсий и обучающих программ.	Свободный доступ

VI. СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

№ п/п	Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
	http://www.biblioclub.ru	Электронно-библиотечная система (ЭБС) Университетская библиотека онлайн	Регистрация через любой университетский компьютер. В дальнейшем индивидуальный неограниченный доступ из любой точки, в которой имеется доступ к сети Интернет
2.	www.elibrary.ru	Российский информационный портал в области науки, технологии, медицины и образования	Свободный доступ
3.	www.iprbookshop.ru	Электронно-библиотечная система (ЭБС)	Доступ возможен с любого компьютера сети ЕГУ или с домашних компьютеров после однократной саморегистрации с любого компьютера университета.

VII. ЛИЦЕНЗИОННОЕ И СВОБОДНО РАСПРОСТРАНЯЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

При реализации учебной дисциплины применяется следующее лицензионное и свободно распространяемое программное обеспечение:

- Microsoft Windows;
- Microsoft Office;

- LibreOffice и др.
- Android Studio

VIII. ОБОРУДОВАНИЕ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБУЧЕНИЯ, НЕОБХОДИМЫЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Учебные занятия проводятся в аудиториях, укомплектованных специализированной мебелью, в том числе стационарными или переносными техническими средствами обучения (проектор, экран, компьютер/ноутбук).

Лабораторные занятия, групповые и индивидуальные консультации, текущая и промежуточная аттестации проводятся в специализированных классах, оснащенных автоматизированными рабочими местами с компьютерами.

Самостоятельная работа проводится в кабинетах, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Предусмотрены помещения для хранения и профилактического обслуживания учебного оборудования.