

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Усынин Максим Валерьевич  
Должность: Ректор  
Дата подписания: 15.06.2015 15:44:17  
Уникальный программный ключ:  
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

Частное образовательное учреждение высшего образования  
**«Международный Институт Дизайна и Сервиса»**  
**(ЧОУВО МИДиС)**

Кафедра математики и информатики

**ФОНД**  
**ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО**  
**КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ**  
**АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**  
**РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ**

Направление подготовки: 09.03.03 Прикладная информатика  
Направленность (профиль): Проектирование и разработка веб  
и мобильных приложений  
Квалификация выпускника: бакалавр

Автор-составитель: Чеботарев С.С.

**СОДЕРЖАНИЕ**

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	3
2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	4
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	7
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	15

## 1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Разработка мобильных приложений» направлен на формирование следующих компетенций:

Код и наименование компетенции выпускника	Код и наименование индикатора достижения компетенций
ПК-3 Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	ПК-3.1. Выполнять действия разработке прототипов информационных систем, мобильных и веб приложений ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и веб приложений в соответствии с требованиями заказчика. ПК 3.3. Владеть инструментами и методами разработки и тестирования баз данных информационных систем
ПК-4 Способен проектировать информационные ресурсы (веб, мобильных приложений) составлять формализованные описания решений, поставленных задач, в соответствии с требованиями, принятыми в организации нормативных документов	ПК-4.1 Составлять формализованные описания решений поставленных задач в соответствии с требованиями, принятыми в организации нормативных документов, выполнять действия по проектированию структур баз данных и дизайну программных интерфейсов. ПК-4.2. Использовать существующие типовые решения и шаблоны информационных ресурсов (веб, мобильных приложений), применять методы и средства проектирования и дизайна информационных ресурсов, баз данных и программных интерфейсов. ПК-4.3. Применять типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке информационных ресурсов (веб, мобильных приложений)
ПК-5 Способен разрабатывать мобильные приложения	ПК-5.1 Осуществлять деятельность по разработке и отладке мобильных приложений. ПК-5.2 Работать со стандартными сервисами платформ и со встроенными устройствами для получения данных, использовать технологии для работы с различными протоколами обмена данными. ПК-5.3 Применять программные средства, технологии и платформы для разработки мобильных приложений, знать основы информационной безопасности.

№ п/п	Код компетенции	Наименование компетенции	Этапы формирования компетенций
1.	ПК-3	Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	<p><i>1 Этап – Знать:</i> ПК-3.1. Способы разработки прототипов информационных систем, мобильных и веб приложений</p> <p><i>2 Этап – Уметь:</i> ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и веб приложений в соответствии с требованиями заказчика</p>

					3 Этап – Владеть: ПК-3.3. Инструментами и методами разработки и тестирования баз данных информационных систем
2.	ПК-4	Способен проектировать информационные ресурсы (веб, мобильных приложений) составлять формализованные описания решений, поставленных задач, в соответствии с требованиями, принятыми в организации нормативных документов			1 Этап – Знать: ПК-4.1. Формализованные способы описания решений поставленных задач в соответствии с требованиями, принятыми в организации нормативных документов; технологии проектирования структур баз данных и дизайна программных интерфейсов
					2 Этап – Уметь: ПК-4.2. Использовать существующие типовые решения и шаблоны информационных ресурсов (веб, мобильных приложений); применять методы и средства проектирования и дизайна информационных ресурсов, баз данных и программных интерфейсов
					3 Этап – Владеть: ПК-4.3. Способами применения типовых решений, библиотек программных модулей, шаблонов, классов объектов, используемых при разработке информационных ресурсов (веб, мобильных приложений)
3.	ПК-5	Способен разрабатывать мобильные приложения			1 Этап – Знать: ПК-5.1. Технологии разработки и отладки мобильных приложений
					2 Этап – Уметь: ПК-5.2. Работать со стандартными сервисами платформ и со встроенными устройствами для получения данных, использовать технологии для работы с различными протоколами обмена данными
					3 Этап – Владеть: ПК-5.3. Программными средствами, технологиями для разработки мобильных приложений, основами информационной безопасности
2.	ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ				
№ п/п	Код компетенции	Наименование компетенции	Критерии оценивания компетенций на различных этапах формирования		Шкала оценивания
1.	ПК-3	Способен проектировать	1 Этап – Знать: ПК-3.1. Способы разработки прототипов		<b>Зачет</b> «ЗАЧТЕНО»: 1. Усвоение программного

		и разрабатывать информационные системы в соответствии с требованиями заказчика	<p>информационных систем, мобильных и веб приложений</p> <p><i>2 Этап – Уметь:</i> ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и веб приложений в соответствии с требованиями заказчика</p> <p><i>3 Этап – Владеть:</i> ПК-3.3. Инструментами и методами разработки и тестирования баз данных информационных систем</p>	<p>материала.</p> <p>2. Знание сущности основных категорий и понятий.</p> <p>3. Выполнение самостоятельной работы за семестр.</p> <p>4. Точность и обоснованность выводов.</p> <p>5. Точные, полные и логичные ответы на дополнительные вопросы.</p> <p><b>«НЕ ЗАЧТЕНО»:</b></p> <p>1. Незнание значительной части программного материала.</p> <p>2. Невыполнение самостоятельной работы за семестр.</p>
2	ПК-4	Способен проектировать информационные ресурсы (веб, мобильных приложений) составлять формализованные описания решений, поставленных задач, в соответствии с требованиями, принятыми в организации нормативных документов	<p><i>1 Этап – Знать:</i> ПК-4.1. Формализованные способы описания решений поставленных задач в соответствии с требованиями, принятыми в организации нормативных документов; технологии проектирования структур баз данных и дизайна программных интерфейсов</p> <p><i>2 Этап – Уметь:</i> ПК-4.2. Использовать существующие типовые решения и шаблоны информационных ресурсов (веб, мобильных приложений); применять методы и средства проектирования и дизайна информационных ресурсов, баз данных и программных интерфейсов</p> <p><i>3 Этап – Владеть:</i> ПК-4.3. Способами применения типовых решений, библиотек программных модулей, шаблонов, классов объектов, используемых при разработке</p>	<p>3. Грубые ошибки при выполнении самостоятельной работы.</p> <p>4. Неумение выделить главное, сделать выводы и обобщения.</p> <p>5. Неправильные ответы на дополнительные вопросы.</p> <p><b>Экзамен</b> <b>«ОТЛИЧНО»</b></p> <p>1. Глубокое и прочное усвоение программного материала.</p> <p>2. Знание пакетов прикладных программ.</p> <p>3. Знание основных принципов построения пакетов прикладных программ.</p> <p>4. Знание основных задач прикладных программ.</p> <p>5. Свободное владение пакетами прикладных программ.</p> <p>6. Точность и обоснованность выводов.</p> <p>7. Безошибочное выполнение практического задания.</p> <p>8. Точные, полные и логичные ответы на дополнительные вопросы.</p> <p><b>«ХОРОШО»</b></p> <p>1. Хорошее знание программного материала.</p> <p>2. Недостаточно полное изложение теоретического</p>

			информационных ресурсов (веб, мобильных приложений)	вопроса экзаменационного билета.
3	ПК-5	Способен разрабатывать мобильные приложения	<i>1 Этап – Знать:</i> ПК-5.1. Технологии разработки и отладки мобильных приложений	3. Наличие незначительных неточностей в употреблении терминов, классификаций.
			<i>2 Этап – Уметь:</i> ПК-5.2. Работать со стандартными сервисами платформ и со встроенными устройствами для получения данных, использовать технологии для работы с различными протоколами обмена данными	4. Знание основных пакетов прикладных программ. 5. Неполнота представленного иллюстративного материала. 6. Точность и обоснованность выводов. 7. Логичное изложение вопроса, соответствие изложения научному стилю.
			<i>3 Этап – Владеть:</i> ПК-5.3. Программными средствами, технологиями для разработки мобильных приложений, основами информационной безопасности	8. Негрубая ошибка при выполнении практического задания. 9. Правильные ответы на дополнительные вопросы.
				<p>«УДОВЛЕТВОРИТЕЛЬНО»</p> <p>1. Поверхностное усвоение программного материала. 2. Недостаточно полное изложение теоретического вопроса экзаменационного билета. 3. Затруднение в приведении примеров, подтверждающих теоретические положения. 4. Наличие неточностей в употреблении терминов, классификаций. 5. Неумение четко сформулировать выводы. 6. Отсутствие навыков научного стиля изложения. 7. Грубая ошибка в практическом задании. 8. Неточные ответы на дополнительные вопросы.</p> <p>«НЕУДОВЛЕТВОРИТЕЛЬНО»</p> <p>1. Незнание значительной части программного материала. 2. Неспособность привести примеры пакетов прикладных программ 3. Неумение выделить главное, сделать выводы и обобщения. 4. Грубые ошибки при выполнении практического задания.</p>

				5. Неправильные ответы на дополнительные вопросы.
--	--	--	--	---

**3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

### **1 ЭТАП – ЗНАТЬ**

#### **Устный опрос по темам:**

- Тема 1. Базовые типы и переменные
- Тема 2. Функции
- Тема 3. Классы и объекты
- Тема 4. Наследование
- Тема 5. Полиморфизм
- Тема 6. Работа с данными
- Тема 7. Безопасность кода
- Тема 8. Структуры данных
- Тема 9. Универсальный код
- Тема 10. Лямбда выражения
- Тема 11. Асинхронность
- Тема 12. Реализация UI на мобильных устройствах
- Тема 13. Архитектурные компоненты

### **2 ЭТАП – УМЕТЬ**

#### **Практические задания**

- Создайте программу на Kotlin, которая объявляет переменные разных типов (Int, Double, String, Boolean), присваивает им значения и выводит их на экран. При этом, продемонстрируйте разницу между ключевыми словами var и val.
- Создайте консольное приложение на Kotlin, которое предлагает пользователю ввести два числа, затем выполните операции сложения, вычитания, умножения и деления с использованием функций. Выведите результаты на экран.
- Создайте класс "Person" с полями для имени и возраста. Создайте объекты этого класса и выведите информацию о них на экран. Добавьте методы для изменения значений полей.
- Создайте базовый класс "Animal" с полями для имени и возраста. От этого класса унаследуйте классы "Dog" и "Cat", добавив специфичные свойства и методы для каждого животного. Создайте объекты обоих подклассов и продемонстрируйте их использование.
- Создайте абстрактный класс "Shape" с методом "area()", который будет реализован в подклассах "Circle" и "Rectangle". Создайте объекты обоих подклассов и выведите их площади на экран.
- Создайте программу на Kotlin, которая считывает данные из файла JSON и выводит их на экран. Затем измените данные и сохраните их обратно в файл. Исследуйте работу с библиотеками для работы с JSON в Kotlin.

### Кейс-задачи

Кейс-задача 1.

**Задание:**

Исследуйте инструментарий языка программирования Kotlin для работы с массивами и коллекциями. Создайте программу, которая считывает данные из внешнего файла (JSON или XML), а затем выводит их на экран. Объясните, как обрабатывать исключения при работе с файлами.

Кейс-задача 2.

**Задание:**

Изучите возможности Kotlin по работе с функциями из одного выражения (Single-Expression Functions) и функциями высшего порядка (Higher-Order Functions). Создайте программу, которая позволяет пользователю выбрать математическую операцию (сложение, вычитание, умножение, деление) с использованием функции высшего порядка.

Кейс-задача 3.

**Задание:**

Изучите концепцию наследования в Kotlin. Создайте суперкласс "Vehicle" и подклассы, представляющие разные виды транспорта (например, "Car" и "Bicycle"). Реализуйте наследование от суперкласса и переопределение свойств в подклассах.

Кейс-задача 4.

**Задание:**

Изучите полиморфизм в Kotlin. Создайте интерфейс "Drawable" с методом "draw()", который будет реализован в классах "Circle" и "Rectangle". Создайте список объектов типа "Drawable" и вызовите метод "draw()" для каждого объекта.

Кейс-задача 5.

**Задание:**

Изучите работу с абстрактными классами и интерфейсами в Kotlin. Создайте интерфейс "Resizable" с методом "resize()", который будет реализован в классах "ResizableCircle" и "ResizableRectangle". Создайте список объектов типа "Resizable" и вызовите метод "resize()" для каждого объекта.

### 3 ЭТАП – ВЛАДЕТЬ

#### Вопросы к зачету

1. Что такое переменная?
2. Чем отличаются ссылка в переменные и объект в переменной?
3. Чем отличаются явный и не явный механизм объявления переменной?
4. Как происходит запись в переменную, массивы?
5. Как работает компилятор, инструментарий языка программирования?
6. Зачем создаются функций?
7. Как происходит передача функций?
8. Как происходит получение значений из функций?
9. Зачем нужна проверка пользовательского ввода?
10. Дайте определение классов, свойства и функции в классе.
11. Опишите блоки инициализации.
12. Зачем необходима гибкая инициализация свойств?

13. Опишите цели наследования.
14. Опишите цели полиморфизма.
15. Назовите свойства интерфейса.
16. Опишите основные этапы подключения к БД
17. Опишите основные этапы работы с сетью

### **Вопросы к экзамену**

1. Какие базовые типы данных поддерживает Kotlin?
2. Чем отличается val от var при объявлении переменных?
3. Какой тип данных следует использовать для хранения целых чисел?
4. Что такое nullable-переменные и как их объявить?
5. Какие операторы доступны для числовых типов данных?
6. Как объявить функцию в Kotlin?
7. Чем отличаются именованные и позиционные аргументы функции?
8. Какие виды функций поддерживаются в Kotlin?
9. Как объявить функцию с параметром по умолчанию?
10. Что такое лямбда-выражения и как их использовать в Kotlin?
11. Как объявить класс в Kotlin?
12. Что такое конструктор и какие виды конструкторов существуют?
13. Как создать экземпляр класса (объект) в Kotlin?
14. Какие особенности имеют объекты и классы данных (data class) в Kotlin?
15. Что такое наследование классов и как его реализовать в Kotlin?
16. Как объявить наследование между классами в Kotlin?
17. Что такое ключевое слово super и как оно используется при наследовании?
18. Какие методы и свойства можно переопределить в дочерних классах?
19. Как реализовать интерфейсы в Kotlin и чем они полезны?
20. Как реализовать абстрактные классы и методы?
21. Что такое полиморфизм и как он реализуется в Kotlin?
22. Какие преимущества полиморфизма при работе с классами?
23. Какие виды полиморфизма существуют в Kotlin?
24. Как использовать переопределенные методы для объектов разных классов?
25. Как обработать исключения и ошибки с помощью полиморфизма?
26. Какие структуры данных используются для хранения коллекций в Kotlin?
27. Как добавить элемент в список (List) в Kotlin?
28. Как удалить элемент из множества (Set) в Kotlin?
29. Как выполнить операции фильтрации и отображения на коллекции?
30. Какие встроенные функции Kotlin облегчают работу с данными?

### **Комплект тестовых вопросов**

1. Какой тип данных в Kotlin используется для хранения целых чисел?  
Int  
Char  
Double  
Какой оператор используется для присвоения значения переменной?  
&  
!  
=  
2. Что такое null-безопасность в Kotlin?

Это возможность использовать null значения в коде

!Это механизм, который позволяет избежать NullPointerException

Это способность переменных сохранять null значения

3. Какой тип данных используется для хранения дробных чисел?

Float

Char

!Double

4. Что такое переменная в Kotlin?

Это объект, который может хранить значение

!Это именованное значение, которое может изменяться в процессе выполнения программы

Это тип данных, который используется для хранения значений

Что такое функция в Kotlin?

Это способность переменных сохранять null значения

!Это именованный блок кода, который может принимать аргументы и возвращать значения

Это объект, который может хранить значение

5. Как объявить функцию в Kotlin?

```
fun myFunction()
```

```
fun myFunction() {}
```

```
!fun myFunction() {}
```

6. Какой тип данных используется для возвращаемого значения функции?

Boolean

!Любой тип данных, включая пользовательские классы и объекты

String

7. Что такое параметр функции в Kotlin?

Это переменная, которая может быть использована только внутри функции

Это способность переменных сохранять null значения

!Это переменная, которая передается в функцию при вызове и используется внутри функции

8. Может ли функция принимать несколько параметров?

Нет, функция может принимать только один параметр

!Да, функция может принимать несколько параметров, разделенных запятой

Зависит от контекста использования функции

### Тема 3. Классы и объекты

9. Что такое класс в Kotlin?

Это способность переменных сохранять null значения

!Это шаблон для создания объектов, который определяет свойства и методы объекта

Это переменная, которая может быть использована только внутри функции

10. Как создать объект класса в Kotlin?

```
object myObject = new object();
```

```
!val myObject = MyClassName()
```

```
val myObject = new MyClassName()
```

11. Что такое конструктор класса в Kotlin?

Это метод класса, который вызывается при создании объекта

Это способность переменных сохранять null значения

!Это метод класса, который используется для инициализации свойств объекта при его создании

12. Что такое свойство класса в Kotlin?

Это метод класса, который вызывается при создании объекта

!Это переменная, которая принадлежит объекту класса и может быть использована для хранения значений

Это способность переменных сохранять null значения

13. Можно ли наследовать классы в Kotlin?

Нет, наследование классов не поддерживается в Kotlin

!Да, наследование классов поддерживается в Kotlin

Зависит от конкретной реализации класса

#### **Тема 4. Наследование**

14. Что такое наследование в Kotlin?

Это способность переменных сохранять null значения

Это возможность одного объекта использовать свойства другого объекта

!Это механизм, который позволяет создавать новый класс на основе существующего класса с использованием его свойств и методов

15. Как объявить класс наследника в Kotlin?

```
class MyChildClass : MyParentClass {}
```

```
!class MyChildClass : MyParentClass() {}
```

```
class MyChildClass extends MyParentClass {}
```

16. Что такое ключевое слово super в Kotlin?

Это ключевое слово, которое используется для создания нового объекта на основе существующего объекта

Это ключевое слово, которое используется для вызова метода родительского класса из метода наследника

!Это ключевое слово, которое ссылается на родительский класс или его конструктор

17. Что такое переопределение методов в Kotlin?

Это возможность использовать методы родительского класса без изменений

!Это возможность изменять реализацию методов родительского класса в классе наследнике

Это возможность создавать новые методы в классе наследнике

18. Каким образом можно запретить наследование класса в Kotlin?

Нельзя запретить наследование класса в Kotlin

!С помощью ключевого слова final перед объявлением класса

С помощью ключевого слова private перед объявлением класса

#### **Тема 5. Полиморфизм**

19. Что такое полиморфизм в Kotlin?

Это возможность использовать одно имя для нескольких методов с разными параметрами или реализацией

Это возможность создавать объекты разных типов с помощью одного конструктора

!Это возможность использовать объекты разных типов как если бы они были одного типа

20. Каким образом можно использовать полиморфизм в Kotlin?

С помощью ключевого слова polymorphism перед объявлением метода

!С помощью переопределения методов родительского класса в классе наследнике

С помощью ключевого слова polymorphism перед объявлением класса

21. Что такое динамическое связывание в Kotlin?

Это возможность использовать методы родительского класса без изменений

Это возможность использовать одно имя для нескольких методов с разными параметрами или реализацией

!Это механизм, который позволяет определить тип объекта во время выполнения программы и вызвать соответствующий метод

22. Каким образом можно проверить тип объекта в Kotlin?

С помощью ключевого слова typeof

С помощью ключевого слова checkType

!С помощью оператора is

23. Что такое абстрактный класс в Kotlin?

Это класс, который нельзя использовать для создания объектов

!Это класс, который содержит абстрактные методы и может быть использован только как базовый класс для других классов

Это класс, который содержит только статические методы

24. Каким образом можно хранить данные в Kotlin?

С помощью баз данных

С помощью файлов на диске

!С помощью переменных и структур данных

25. Какие типы данных можно хранить в массиве в Kotlin?

Только числа и строки

Только числа и булевы значения

!Любые типы данных, включая пользовательские классы и объекты

26. Что такое коллекции в Kotlin?

Это способность переменных сохранять null значения

Это способ хранения данных на диске

!Это структуры данных, которые позволяют хранить и управлять группами элементов

27. Какие типы коллекций поддерживает Kotlin?

Только списки и массивы

Только словари и множества

!Списки, массивы, словари и множества

28. Каким образом можно получить доступ к элементу коллекции по индексу в Kotlin?

С помощью оператора .

С помощью оператора ->

!С помощью оператора []

### Тема 7. Безопасность кода

29. Что такое исключение в Kotlin?

Это способность переменных сохранять null значения

Это ошибка компиляции кода

!Это ошибка выполнения программы, которая может привести к остановке программы или потере данных

30. Каким образом можно обработать исключение в Kotlin?

С помощью операторов if/else

С помощью ключевого слова catch и блока try-catch-finally

!С помощью операторов switch/case

31. Что такое безопасный вызов оператора (safe call operator) в Kotlin?

Это оператор, который позволяет вызывать метод без проверки на null значение объекта

!Это оператор, который позволяет вызывать метод только если объект не является null значением

Это оператор, который позволяет вызывать метод только если объект является null значением

32. Что такое Elvis operator в Kotlin?

Это оператор, который позволяет вызывать метод без проверки на null значение объекта

!Это оператор, который позволяет задавать значение по умолчанию для переменной, если она является null значением

Это оператор, который позволяет вызывать метод только если объект является null значением

33. Что такое стек (Stack) в Kotlin?

Это структура данных, которая хранит элементы последним пришел - первым ушел (Last In First Out)

Это структура данных, которая хранит элементы первым пришел - первым ушел (First In First Out)

!Это структура данных, которая позволяет добавлять и удалять элементы только с одного конца (вершины)

34. Каким образом можно добавить элемент в стек (Stack) в Kotlin?

С помощью метода add()

С помощью метода insert()

!С помощью метода push()

35. Каким образом можно удалить элемент из стек (Stack) в Kotlin?

С помощью метода remove()

С помощью метода delete()

!С помощью метода pop()

36. Что такое очередь (Queue) в Kotlin?

Это структура данных, которая хранит элементы последним пришел - первым ушел (Last In First Out)

!Это структура данных, которая хранит элементы первым пришел - первым ушел (First In First Out) -Это структура данных, которая позволяет добавлять и удалять элементы только с одного конца (вершины)

37. Что такое универсальный код (Unicode)?

Это система кодирования символов на основе двоичной системы счисления.

Универсальный код не используется в программировании.

!Это международный стандарт кодирования символов всех письменных языков.

38. Какие символы могут быть закодированы с помощью универсального кодирования (Unicode)?

Только символы латинского алфавита.

!Символы всех письменных языков.

Только символы кириллицы.

39. Каким образом можно представить символ Unicode в программном коде на языке Kotlin?

С помощью последовательности байт.

!С помощью шестнадцатеричного кода символа.

С помощью двоичного кода символа.

40. Каким образом можно получить символ Unicode по его шестнадцатеричному коду в программном коде на языке Kotlin?

С помощью оператора typeof.

!С помощью оператора unicode.

С помощью оператора hex.

41. Что такое эскейп последовательности (Escape sequence) в программировании на языке Kotlin?

Это последовательность символов, которая используется для перехода к следующей строке.

!Это последовательность символов, которая используется для представления специальных символов, таких как кавычки или символ переноса строки.

Это последовательность символов, которая используется для представления чисел с плавающей точкой.

### **Тема 11. Асинхронность**

42. Что такое асинхронность в Kotlin?

Это способность приложения выполнять операции последовательно

!Это способность приложения выполнять операции параллельно

Это способность приложения выполнять операции с задержкой

43. Какой класс используется для выполнения асинхронных операций в Kotlin?

!AsyncTask

Handler

Runnable

44. Что такое корутины (coroutines) в Kotlin?

Это класс для выполнения асинхронных операций

Это библиотека для работы с базами данных

!Это легковесный механизм для выполнения асинхронных операций

Как объявить корутину в Kotlin?

```
!fun main() = runBlocking { ... }
```

```
fun main() { ... }
```

```
val myCoroutine = Coroutine()
```

45. Какой оператор используется для приостановки выполнения корутины в Kotlin?

!delay()

pause()

sleep()

### Тема 12. Реализация UI на мобильных устройствах

46. Что такое View в Android?

Это класс для работы с базами данных

Это библиотека для создания пользовательского интерфейса

!Это базовый класс для всех элементов пользовательского интерфейса

47. Какие основные компоненты пользовательского интерфейса в Android?

!TextView, EditText, Button, ImageView, ProgressBar

Database, Network, Service, Broadcast Receiver, Content Provider

Activity, Fragment, Service, BroadcastReceiver, Content Provider

48. Что такое Layout в Android?

Это класс для работы с базами данных

Это библиотека для создания пользовательского интерфейса

!Это файл XML, который определяет расположение и свойства элементов пользовательского интерфейса

49. Как добавить элементы пользовательского интерфейса в Layout в Android?

!С помощью тегов XML

С помощью функции addView() в коде приложения

С помощью функции setContentView() в коде приложения

50. Как обрабатывать события нажатия на элементы пользовательского интерфейса в Android?

С помощью функции onClickListener() в коде приложения

С помощью функции setOnClickListener() в коде приложения

!С помощью атрибута onClick в XML файле Layout

### Тема 13. Архитектурные компоненты

51. Что такое архитектурные компоненты в Android?

Это библиотека для работы с базами данных

!Это набор компонентов и библиотек, которые помогают создавать хорошо структурированные приложения с четким разделением обязанностей

Это библиотека для создания пользовательского интерфейса

52. Какой компонент используется для хранения и управления данными в Android?

Activity

Fragment

!ViewModel

53. Какой компонент используется для связывания данных и пользовательского интерфейса в Android?

!Data Binding  
View Binding  
Layout Binding

54. Что такое LiveData в Android?

!Это класс, который позволяет создавать объекты, которые могут быть наблюдаемыми и автоматически обновляться при изменении значений

Это класс для работы с базами данных

Это класс для создания пользовательского интерфейса

55. Какой компонент используется для работы с базами данных в Android?

!Room

SQLiteOpenHelper

Content Provider

#### 4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

##### 1 ЭТАП – ЗНАТЬ

##### Критерии оценки устных ответов студентов

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно.
«хорошо»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно, но допускает 1-2 ошибки, которые исправляет сам.
«удовлетворительно»	знает и понимает основные положения по содержанию задания; излагает материал неполно, но допускает неточности в определении понятий или формулировке правил; не достаточно глубоко и доказательно обосновать свои суждения иллюстрирует ответ примерами только по предложенной литературе; излагает материал непоследовательно и допускает 3-4 ошибки.
«неудовлетворительно»	допускает существенные ошибки в формулировке определений и правил, искажающие их смысл; излагает материал непоследовательно, неуверенно и не по существу задания; допускает существенные ошибки, не позволяющие раскрыть смысл задания, являющиеся серьезным препятствием к успешному овладению следующим материалом.

**2 ЭТАП – УМЕТЬ****Критерии оценивания выполнения практических заданий**

Оценка	Критерии
«отлично»	Работа выполнена и оформлена верно; использован оптимальный метод решения
«хорошо»	Работа выполнена и оформлена в целом верно с небольшими ошибками
«удовлетворительно»	Работа выполнена и оформлена с ошибками; использован неполный метод решения
«неудовлетворительно»	Работа не выполнена в срок

**Критерии оценивания работы с кейс-задачами**

Оценка	Критерии
«отлично»	разработана архитектура классов; разработано приложение; на защите кейс-задачи были получены верные ответы на все дополнительные вопросы
«хорошо»	разработана архитектура классов; разработано приложение; на защите кейс-задачи при ответах на вопросы были допущены ошибки
«удовлетворительно»	разработана архитектура классов; разработано приложение, но в приложении имеются ошибки и недоработки; на защите кейс-задачи при ответах на вопросы были допущены ошибки
«неудовлетворительно»	разработана архитектура классов; не было разработано приложение

**3 ЭТАП – ВЛАДЕТЬ****Критерии оценивания знаний на зачете**

«ЗАЧТЕНО»:

1. Усвоение программного материала.
2. Знание сущности основных категорий и понятий.
3. Выполнение самостоятельной работы за семестр.
4. Точность и обоснованность выводов.
5. Точные, полные и логичные ответы на дополнительные вопросы.

«НЕ ЗАЧТЕНО»:

1. Незнание значительной части программного материала
2. Невыполнение самостоятельной работы за семестр.
3. Грубые ошибки при выполнении самостоятельной работы.
4. Неумение выделить главное, сделать выводы и обобщения.
5. Неправильные ответы на дополнительные вопросы.

**Критерии оценивания знаний на экзамене**

Оценка	Критерии
«отлично»	<ol style="list-style-type: none"> <li>1. Глубокое и прочное усвоение программного материала.</li> <li>2. Знание пакетов прикладных программ.</li> <li>3. Знание основных принципов построения пакетов прикладных программ.</li> <li>4. Знание основных задач прикладных программ.</li> <li>5. Свободное владение пакетами прикладных программ.</li> <li>6. Точность и обоснованность выводов.</li> <li>7. Безошибочное выполнение практического задания.</li> </ol>

	8. Точные, полные и логичные ответы на дополнительные вопросы.
«хорошо»	<ol style="list-style-type: none"> <li>1. Хорошее знание программного материала.</li> <li>2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.</li> <li>3. Наличие незначительных неточностей в употреблении терминов, классификаций.</li> <li>4. Знание основных пакетов прикладных программ.</li> <li>5. Неполнота представленного иллюстративного материала.</li> <li>6. Точность и обоснованность выводов.</li> <li>7. Логичное изложение вопроса, соответствие изложения научному стилю.</li> <li>8. Негрубая ошибка при выполнении практического задания.</li> <li>9. Правильные ответы на дополнительные вопросы.</li> </ol>
«удовлетворительно»	<ol style="list-style-type: none"> <li>1. Поверхностное усвоение программного материала.</li> <li>2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.</li> <li>3. Затруднение в приведении примеров, подтверждающих теоретические положения.</li> <li>4. Наличие неточностей в употреблении терминов, классификаций.</li> <li>5. Неумение четко сформулировать выводы.</li> <li>6. Отсутствие навыков научного стиля изложения.</li> <li>7. Грубая ошибка в практическом задании.</li> <li>8. Неточные ответы на дополнительные вопросы.</li> </ol>
«неудовлетворительно»	<ol style="list-style-type: none"> <li>1. Незнание значительной части программного материала.</li> <li>2. Неспособность привести примеры пакетов прикладных программ</li> <li>3. Неумение выделить главное, сделать выводы и обобщения.</li> <li>4. Грубые ошибки при выполнении практического задания.</li> <li>5. Неправильные ответы на дополнительные вопросы.</li> </ol>

### Критерии оценивания результатов теста

Полная версия тестовых вопросов содержится в электронно-информационной системе вуза. Студенты проходят тестирование в компьютерном классе. Оценка успешности прохождения теста определяется следующей сеткой: от 0% до 29% – «неудовлетворительно», от 30% до 59% – «удовлетворительно»; 60% – 79 % – «хорошо»; 80% -100% – «отлично».