Отчёт по практике

**«Разработка приложений для мобильных устройств»**

Задание 1

Разработка экрана старта игры и логики игрового персонажа

Выполнил:

Студент группы 8B21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Д. Алексейчик

# Задание

Разработайте стартовый экран приложения "Кошки-мышки" и базовую логику для персонажа "Мышь".

1. Создайте Activity MainActivity, которая будет отображать стартовый экран.
2. На экране разместите:

* Текстовое поле (TextView) с названием игры "Кошки-Мышки".
* Кнопку (Button) с текстом "Начать игру".
* При нажатии на кнопку должно появляться короткое текстовое сообщение (Toast) "Игра начинается!".

1. В отдельном Kotlin-файле создайте класс Mouse (Мышь).

* Класс должен иметь свойства: name (имя, строка) и speed (скорость, целое число). Оба свойства должны быть объявлены в первичном конструкторе.
* Создайте функцию getMouseInfo(), которая возвращает строку вида: "Мышь по имени [имя] бежит со скоростью [скорость]".
* Учтите null-безопасность: свойство name должно быть инициализировано пустой строкой по умолчанию.

# ****Исходные данные:****

1. Проектирование UI: Макет экрана activity\_main.xml должен содержать вертикальный LinearLayout или ConstraintLayout с указанными элементами.
2. Программирование на Kotlin: Класс Mouse должен быть реализован в файле Mouse.kt.
3. Работа с Android Components: Код для обработки нажатия кнопки и показа Toast должен быть в MainActivity.kt.
4. Тестирование: Написать модульный тест (JUnit) для класса Mouse, который проверяет корректность работы функции getMouseInfo().

# Ход работы

1. Создайте Activity MainActivity, которая будет отображать стартовый экран.
2. На экране разместите:

* Текстовое поле (TextView) с названием игры "Кошки-Мышки".
* Кнопку (Button) с текстом "Начать игру".
* При нажатии на кнопку должно появляться короткое текстовое сообщение (Toast) "Игра начинается!".

MainActivity.kt

class MainActivity : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 // устанавливаем обработчик событий на кнопку  
 val btnStart: Button = findViewById(R.id.*btnStart*)  
 btnStart.setOnClickListener **{** //сообщение Toast при нажатии на кнопку  
 Toast.makeText(this, "Игра начинается!", Toast.*LENGTH\_SHORT*).show()  
 **}** }  
}

Класс MainActivity представляет главную активность приложения. В методе onCreate() инициализируется пользовательский интерфейс и устанавливается обработчик нажатия на кнопку «Начать игру», который при клике выводит краткое всплывающее уведомление типа Toast.

Toast – краткое всплывающее уведомление в Android, которое появляется на короткое время в нижней части экрана и автоматически исчезает.

activity\_main.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/ll\_root"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:padding="24dp">  
  
 <TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Кошки-Мышки"  
 android:textSize="32sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="32dp" />  
  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Начать игру" />  
  
</LinearLayout>

Описание пользовательского интерфейса главного меню, состоящего из текста с названием игры “Кошки-Мышки” и кнопки “Начать игру”.

1. В отдельном Kotlin-файле создайте класс Mouse (Мышь).

* Класс должен иметь свойства: name (имя, строка) и speed (скорость, целое число). Оба свойства должны быть объявлены в первичном конструкторе.
* Создайте функцию getMouseInfo(), которая возвращает строку вида: "Мышь по имени [имя] бежит со скоростью [скорость]".
* Учтите null-безопасность: свойство name должно быть инициализировано пустой строкой по умолчанию.

Mouse.kt

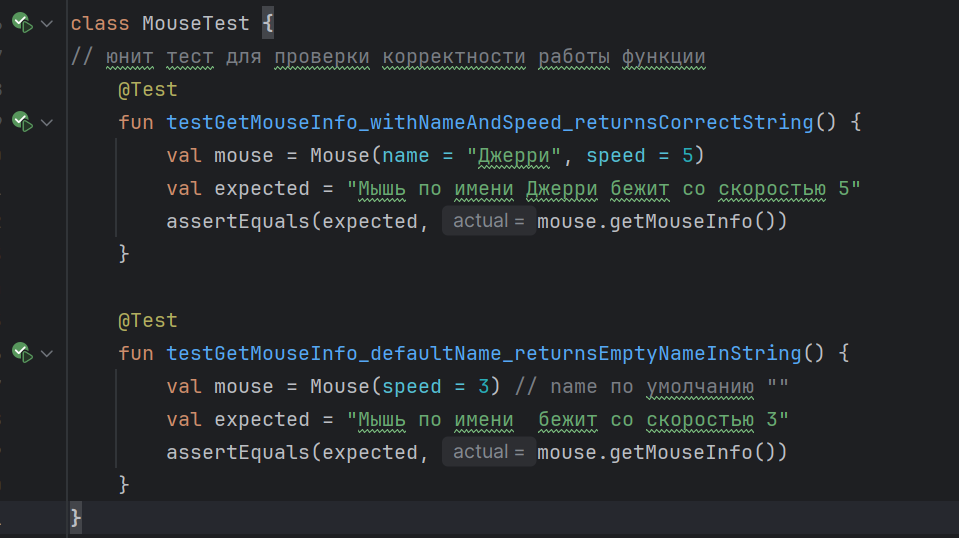
// класс для мыши  
data class Mouse(  
 val name: String = "", // имя  
 val speed: Int // скорость мыши  
) {  
 fun getMouseInfo(): String { // отображение информации о мыши  
 return "Мышь по имени $name бежит со скоростью $speed"  
 }  
}

Класс Mouse содержит параметры игрока – имя и скорость. Функция getMouseInfo возвращает строку "Мышь по имени [имя] бежит со скоростью [скорость]".

Тестирование:

В файле MouseTest.kt представлен юнит-тест, позволяющий проверить правильность работы класса Mouse.

Юнит-тест (unit test) – автоматизированный тест, который проверяет корректность работы отдельного модуля кода (функции или метода), изолируя его от остальных частей приложения.

Рисунок 1. Успешно пройденные юнит-тесты

# Результат работы

Ссылка на GitHub: https://github.com/AlekJulia/Alekseychik\_MobileApplicationDevelopment.git

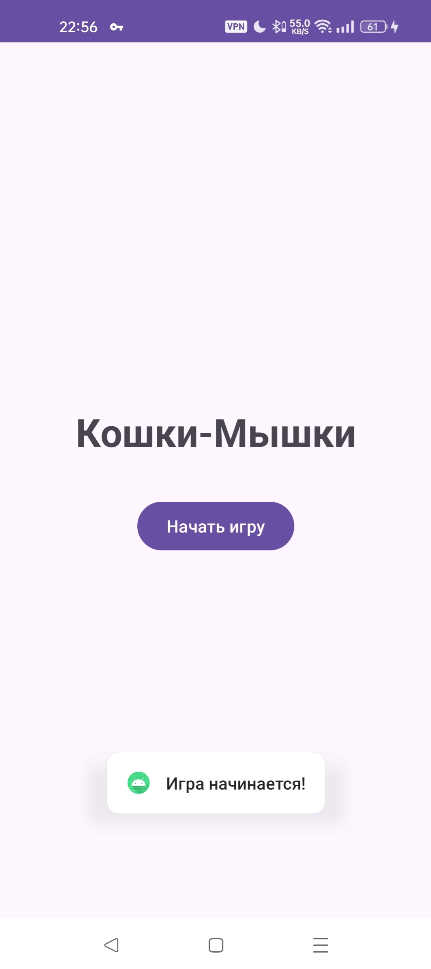


Рисунок 2. Вывод сообщения при нажатии на кнопку “Начать игру”