Государственное бюджетное профессиональное образовательное учреждение

«Нижегородский Губернский колледж»

**Юнит-тесты**

**Клиентская часть приложения CodeBlitz**

Группа: 41П

Студент: Фадеев А. В.

Нижний Новгород

2024

# ОСНОВНАЯ ЧАСТЬ

## Описание юнит-тестов

Все юнит-тесты были написаны на языке Kotlin.

Таблица 1 – Юнит-тесты

| Номер | Название | Описание | Ожидаемый результат |
| --- | --- | --- | --- |
| 1 | Проверка обновления данных регистрации | Проверяет, что после вставки данных в LoginViewModel они корректно сохраняются | Сохранились те же данные, что и были введены |
| 2 | Проверка доступности кнопки входа при вводе данных в поля | Проверяет, что поле, отвечающее за состояние кнопки, изменило свое состояние при вводе логина и пароля в поля | Поле изменило значение на true |
| 3 | Проверка потока навигации на страницу регистрации | Проверяет, что значение потока навигации после вызова функции навигации стало содержать в себе путь на страницу регистрации | Значение потока навигации равно объекту, содержащему путь к регистрации |
| 4 | Проверка функции подсчета строк | Проверяет корректный формат строки с номерами строчек редактора | Формат строки соответствует количеству строк введенного текста |
| 5 | Проверка записи описания задачи в поле | Проверяет, что описание задачи корректно записалось в поле ViewModel | В поле ViewModel записалось введенное значение |

Ниже представлен листинг всех юнит-тестов:

Листинг 1 – Юнит-тесты

|  |
| --- |
| package com.example.codeblitz  import androidx.compose.ui.text.input.TextFieldValue import com.example.codeblitz.domain.AddTaskViewModel import com.example.codeblitz.domain.LoginViewModel import com.example.codeblitz.domain.RegisterViewModel import com.example.codeblitz.domain.navigation.Routes import com.example.codeblitz.model.LoginData import com.example.codeblitz.view.MainActivity.components.countAndFormatNewLines import kotlinx.coroutines.test.runTest import org.junit.Assert.assertEquals import org.junit.Test  */\*\*  \* Юнит-тесты  \*/* class CodeBlitzUnitTests {  */\*\*  \* Проверка обновления данных регистрации  \*/* @Test  fun testUpdateRegisterData() {  val viewModel = RegisterViewModel()  *assert*(viewModel.registerData.password.*isEmpty*())  *assert*(viewModel.registerData.login.*isEmpty*())   viewModel.updateRegisterData(  newRegisterData = LoginData("123", "1234")  )   assertEquals("1234", viewModel.registerData.password)  assertEquals("123", viewModel.registerData.login)  }   */\*\*  \* Проверка доступности кнопки входа при вводе данных в поля, отвечающие за логин и пароль  \*/* @Test  fun testIsLoginButtonActive() {  val viewModel = LoginViewModel()  *assert*(!viewModel.isButtonEnabled)   viewModel.updateLoginData(  newLoginData = LoginData("123", "1234")  )   *assert*(viewModel.isButtonEnabled)  }   */\*\*  \* Проверка потока навигации на страницу регистрации  \*/* @Test  fun testToRegisterNavigationFlow() = *runTest* **{** val viewModel = LoginViewModel()  viewModel.navigateToRegister()   assertEquals(Routes.Register, viewModel.navigationStateFlow.value)  **}** */\*\*  \* Проверка функции подсчета строк  \*/* @Test  fun testCountLines() {  val string = """  # Example usage:  numbers = list(range(1, 101)) #Test with a small list  start\_time = time.time()  result = incredibly\_inefficient\_sum(numbers) # This will break with a longer list due to recursion depth  end\_time = time.time()  print(f"Sum: {result}")  print(f"Time taken: {end\_time - start\_time:.4f} seconds")     numbers = list(range(1,10)) #Test with a small list that will not cause stack overflow  start\_time = time.time()  result = incredibly\_inefficient\_sum(numbers)  end\_time = time.time()  print(f"Sum: {result}")  print(f"Time taken: {end\_time - start\_time:.4f} seconds")  """.*trimIndent*()   val expected = "1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n11\n12\n13\n14\n15\n16"  val result = *countAndFormatNewLines*(TextFieldValue(string))   assertEquals(expected, result)  }   */\*\*  \* Проверка записи описания задачи в поле  \*/* @Test  fun testTaskDesc() {  val viewModel = AddTaskViewModel()   val expected = """  the most beautiful description  """.*trimIndent*()   viewModel.setTaskDesc(expected)   assertEquals(expected, viewModel.taskDesk)  } } |

Все юнит-тесты были успешно пройдены:

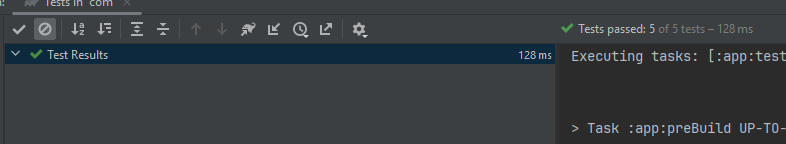


Рисунок 1 – Выполнение юнит-тестов