Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Отчёт по предмету «Программирование мобильных информационных

систем»

по лабораторной работе №1

на тему:

**ОСНОВЫ ЯЗЫКА *KOTLIN***

Проверил      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Усенко Ф.В.

(подпись)

Выполнил      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Косович П. В.,

(подпись) гр.214302

Минск 2024

**Цель работы: Сложная игра «Угадай слово с подсказками».** Изучить основные конструкции языка *Kotlin*, научиться работать с типами данных, операциями ввода-вывода, а также условными операторами. Закрепить полученные знания через выполнение задач, требующих практического применения теории.

**КОД ПРОГРАММЫ**

import java.util.\*  
  
fun main() {  
 do {  
 playGame()  
 println("Хотите сыграть еще раз? (да/нет)")  
 } while (readlnOrNull()?.lowercase(Locale.getDefault()) == "да")  
 println("Спасибо за игру!")  
}  
  
fun playGame() {  
 val words = listOf("kotlin", "java", "python", "swift", "javascript", "scala", "ruby")  
 val levels = mapOf(  
 1 to "Легкий",  
 2 to "Средний",  
 3 to "Трудный"  
 )  
 println("Добро пожаловать в игру 'Угадай слово с подсказками'!")  
 println("Выберите уровень сложности:")  
 for ((key, value) in levels) {  
 println("$key - $value")  
 }  
 val level = readlnOrNull()?.toIntOrNull() ?: 1  
 val maxAttempts = when (level) {  
 1 -> Int.MAX\_VALUE  
 2 -> 7  
 3 -> 5  
 else -> Int.MAX\_VALUE  
 }  
 println("Количество попыток для игры: $maxAttempts")  
 val selectedWord = words.random()  
 var attempts = 0  
 var guessedCorrectly = false  
 var hint = "\_".repeat(selectedWord.length)  
 println("Игра началась! Слово состоит из ${selectedWord.length} букв.")  
  
 while (attempts < maxAttempts && !guessedCorrectly) {  
 println("Попытка ${attempts + 1}. Введите слово:")  
 val userInput = readlnOrNull()?.lowercase(Locale.getDefault()) ?: ""  
 if (userInput == selectedWord) {  
 guessedCorrectly = true  
 break  
 } else {  
 hint = provideHint(userInput, selectedWord, level, hint)  
 attempts++  
 }  
 }  
  
 if (guessedCorrectly) {  
 println("Поздравляем! Вы угадали слово '$selectedWord' за $attempts попыток!")  
 } else {  
 println("К сожалению, вы не смогли угадать слово. Правильное слово было: $selectedWord")  
 }  
}  
  
fun provideHint(userInput: String, selectedWord: String, level: Int, currentHint: String): String {  
 val hint = StringBuilder(currentHint)  
 val minLength = minOf(userInput.length, selectedWord.length)  
  
 for (i in 0 until minLength) {  
 if (userInput[i] == selectedWord[i]) {  
 hint.set(i, userInput[i])  
 }  
 }  
  
 when (level) {  
 1 -> {  
 println("Подсказка: $hint")  
 }  
 2 -> {  
 println("Угаданы ${countMatchingLetters(userInput, selectedWord)} букв, правильные позиции: $hint")  
 }  
 3 -> {  
 println("Угаданы ${countMatchingLetters(userInput, selectedWord)} букв.")  
 }  
 }  
  
 return hint.toString()  
}  
  
fun countMatchingLetters(userInput: String, selectedWord: String): Int {  
 return userInput.zip(selectedWord).count **{** it.first == it.second **}**}

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

**1 Как организован ввод данных с консоли в *Kotlin*?**

Ввод данных от пользователя в *Kotlin* осуществляется с помощью функции *readLine*(), которая возвращает введенную строку. Для преобразования строки в число используются методы *toInt*(), *toDouble*(), и т.д.

**2 Какие типы данных существуют в *Kotlin* для целых чисел?**

*Byte* (8 бит), *Short* (16 бит), *Int* (32 бита), *Long* (64 бита) – типы данных для хранения целых чисел.

**3 Какие операторы используются для сравнения чисел в Kotlin?**

==, != — проверка на равенство и неравенство;

>, < — больше и меньше;

>=, <= — больше или равно, меньше или равно.

**ВЫВОД**

В ходе выполнения лабораторной работы были изучены основные конструкции языка *Kotlin*, удалось научиться работать с типами данных, операциями ввода-вывода, а также условными операторами. Полученные знания были закреплены через выполнение задач, требующих практическое применение теории.