

Практическая работа: Основы Kotlin: переменные и функции

Прежде чем начать

Вы приложили немало усилий, чтобы получить знания об основах программирования на Kotlin, но теперь пришло время применить полученные знания на практике.

Эти упражнения проверяют ваше понимание изученных концепций. Они основаны на реальных примерах использования, с некоторыми из которых вы, вероятно, уже сталкивались.

Следуйте инструкциям, чтобы найти решение для каждой задачи. Если вы застряли, в некоторых упражнениях есть подсказки, которые могут вам помочь.

Рассматривайте решения как один из способов решения задач и не стесняйтесь экспериментировать, как вам удобно. Вы можете решить некоторые упражнения несколькими способами и использовать разные имена для функций и переменных.

Для решения этих заданий рекомендуется использовать Idea.

Необходимые условия

- Знакомство с Idea
- Умение определять и вызывать функции
- Знание основных типов данных Kotlin
- Знание неизменяемых и изменяемых переменных
- Знание функции `println()`

Что вам понадобится

- Компьютер с Idea

1. Напечатайте сообщения

Расскажите своим друзьям, чему вы научились на этом пути.

Можете ли вы написать функцию `main()`, которая печатает эти сообщения на четырех отдельных строках?

Используйте ключевое слово `val`, когда значение не меняется.
Используйте ключевое слово `var`, когда значение может меняться.
Когда вы определяете функцию, вы определяете параметры, которые могут быть ей переданы.
Когда вы вызываете функцию, вы передаете аргументы для параметров.

2. Исправление ошибки компиляции

Эта программа печатает сообщение, уведомляющее пользователя о том, что он получил сообщение в чате от друга.

```
fun main() {  
    println(«Новое сообщение в чате от друга»)  
}
```

Можете ли вы выяснить причину ошибок компиляции в этой программе и исправить их? Используются ли в коде соответствующие символы для обозначения открытия и закрытия строки и аргумента функции?

Подсказка: Вы можете использовать Idea, чтобы запустить код и просмотреть ошибки компиляции.

После исправления ошибок программа должна скомпилироваться без ошибок и вывести такой вывод:

```
Новое сообщение в чате от друга
```

3. Шаблоны строк

Эта программа информирует пользователей о предстоящей акционной распродаже на определенный товар. В ней есть строковый шаблон, который опирается на переменную `discountPercentage` для процентной скидки и переменную `item` для продаваемого товара. Однако в коде присутствуют ошибки компиляции.

```
fun main() {  
    val discountPercentage: Int = 0  
    val предложение: String = «»  
    val item = «Google Chromecast»  
    discountPercentage = 20  
    offer = «Распродажа - скидка до $discountPercentage% на $item! Поторопитесь!»  
  
    println(offer)  
}
```

- Можете ли вы выяснить причину ошибок и исправить их?
- Можете ли вы определить вывод этой программы до того, как запустите код в Idea?

Подсказка: Можете ли вы переназначить значение переменной, доступной только для чтения?

После исправления ошибок программа должна скомпилироваться без ошибок и вывести такой вывод:

Распродажа - Скидка до 20% на Google Chromecast! Торопитесь!

4. Конкатенация строк

Эта программа выводит общее количество участников вечеринки. На вечеринке присутствуют взрослые и дети. Переменная `numberOfAdults` содержит количество взрослых на вечеринке, а переменная `numberOfKids` - количество детей.

```
fun main() {  
    val numberOfAdults = «20»  
    val numberOfKids = «30»  
    val total = numberOfAdults + numberOfKids  
    println(«Общий размер вечеринки составляет: $total»)  
}
```

Шаг 1

- Можете ли вы определить вывод этой программы до того, как запустите код в Idea?
- После того как вы определили вывод, запустите код в Idea и проверьте, совпадает ли ваш вывод с выведенным на экран.

Подсказка: Что произойдет, если использовать оператор `+` для двух строк?

Шаг 2

Код работает и печатает некоторые выходные данные, но в них не отображается общее количество людей, присутствующих на вечеринке.

- Можете ли вы найти проблему в коде и исправить его так, чтобы он выводил этот результат?

Общее количество участников вечеринки составляет: 50

5. Форматирование сообщения

Эта программа отображает общую зарплату, которую сотрудник получает в этом месяце. Общая зарплата делится на две части: переменная `baseSalary`, которую сотрудник получает каждый месяц, и переменная `bonusAmount`, которая является дополнительной премией, начисленной сотруднику.

```
fun main() {  
    val baseSalary = 5000  
    val BonusAmount = 1000  
    val totalSalary = «$baseSalary + $bonusAmount»  
}
```

```
println("Поздравляем с получением бонуса! Вы получите общую сумму $totalSalary  
(дополнительный бонус).")  
}
```

- Можете ли вы определить вывод этого кода до того, как запустите его в Idea?
- Когда вы запустите код в Idea, выведет ли он то, что вы ожидали?

6. Выполнение основных математических операций

В этом упражнении вы напишете программу, которая выполняет основные математические операции и печатает результат.

Шаг 1 Функция `main()` содержит одну ошибку компиляции:

```
fun main() {  
    val firstNumber = 10  
    val secondNumber = 5  
  
    println(«$firstNumber + $secondNumber = $result»)  
}
```

- Можете ли вы исправить ошибку, чтобы программа выводила такой результат?

```
10 + 5 = 15
```

Шаг 2 Код работает, но логика сложения двух чисел находится в переменной `result`, что делает ваш код менее гибким для повторного использования. Вместо этого вы можете вынести операцию сложения в функцию `add()`, чтобы код можно было использовать повторно. Для этого обновите свой код с помощью приведенного ниже кода. Обратите внимание, что теперь в коде появилась новая переменная `val` под названием `thirdNumber` и результат этой новой переменной печатается вместе с `firstNumber`.

```
fun main() {  
    val firstNumber = 10  
    val secondNumber = 5  
    val thirdNumber = 8  
  
    val result = add(firstNumber, secondNumber)  
    val anotherResult = add(firstNumber, thirdNumber)  
  
    println(«$firstNumber + $secondNumber = $result»)  
    println(«$firstNumber + $thirdNumber = $anotherResult»)  
}  
  
// Определите функцию add() под этой строкой
```

- Можете ли вы определить функцию `add()` так, чтобы программа печатала этот вывод?

```
10 + 5 = 15
10 + 8 = 18
```

Шаг 3 Теперь у вас есть многократно используемая функция для сложения двух чисел.

- Можете ли вы реализовать функцию `subtract()` так же, как реализовали функцию `add()`? Измените функцию `main()` так, чтобы она использовала функцию `subtract()`, и вы сможете проверить, что она работает так, как ожидалось.

Подсказка: подумайте о разнице между сложением, вычитанием и другими математическими операциями. Начните работу над кодом решения с этого момента.

7. Параметры по умолчанию

В Gmail есть функция, которая отправляет пользователю уведомление при каждой попытке входа в систему на новом устройстве.

В этом упражнении вы напишете программу, которая выведет на экран сообщение для пользователей с таким шаблоном сообщения:

```
На operatingSystem поступил новый запрос на вход в систему для вашего emailId
аккаунта Google.
```

Вам нужно реализовать функцию, которая принимает параметр `operatingSystem` и параметр `emailId`, строит сообщение в заданном формате и возвращает его.

Например, если функция была вызвана с параметрами «Chrome OS» для операционной системы и «sample@gmail.com» для emailId, она должна вернуть такую строку:

```
Поступил новый запрос на вход в Chrome OS для вашего аккаунта Google
sample@gmail.com.
```

Шаг 1 Можете ли вы реализовать функцию `displayAlertMessage()` в этой программе так, чтобы она печатала выводимое сообщение?

```
fun main() {
    val operatingSystem = «Chrome OS»
    val emailId = «sample@gmail.com»

    println(displayAlertMessage(operatingSystem, emailId))
}
```

```
}  
  
// Определите свой displayAlertMessage() ниже этой строки.
```

Печатает ли ваша программа этот вывод?

В Chrome OS появился новый запрос на вход в аккаунт Google sample@gmail.com.

Шаг 2 Отличная работа! Вы отобрали сообщение. Однако в некоторых случаях вы обнаруживаете, что не можете определить операционную систему пользователя. В таких случаях необходимо указать имя операционной системы как Unknown OS. Вы можете дополнительно оптимизировать код, чтобы не передавать аргумент Unknown OS при каждом вызове функции.

Можете ли вы найти способ оптимизировать код с этой информацией так, чтобы он выводил такой результат?

Поступил новый запрос на вход в Неизвестную ОС для вашего аккаунта Google user_one@gmail.com.

Поступил новый запрос на вход в систему в Windows для вашего аккаунта Google user_two@gmail.com.

Есть новый запрос на вход в систему на Mac OS для вашего аккаунта Google user_three@gmail.com.

Чтобы вывести приведенное выше сообщение, замените реализацию функции `main()` на эту:

```
fun main() {  
    val firstUserEmailId = «user_one@gmail.com»  
  
    // Следующая строка кода предполагает, что вы назвали свой параметр как  
    emailId.  
    // Если вы назвали его по-другому, не стесняйтесь обновить имя.  
    println(displayAlertMessage(emailId = firstUserEmailId))  
    println()  
  
    val secondUserOperatingSystem = «Windows»  
    val secondUserEmailId = «user_two@gmail.com»  
  
    println(displayAlertMessage(secondUserOperatingSystem, secondUserEmailId))  
    println()  
  
    val thirdUserOperatingSystem = «Mac OS»  
    val thirdUserEmailId = «user_three@gmail.com»  
  
    println(displayAlertMessage(thirdUserOperatingSystem, thirdUserEmailId))  
}
```

```
println()  
}
```

8. Шагомер

Шагомер - это электронное устройство, которое подсчитывает количество пройденных шагов. В настоящее время почти все мобильные телефоны, умные часы и фитнес-тренажеры оснащены встроенными шагомерами. Приложение «Здоровье и фитнес» использует встроенные шагомеры для подсчета количества пройденных шагов. Эта функция рассчитывает количество калорий, которые сжигает пользователь, исходя из количества его шагов.

Можете ли вы переименовать функции, параметры функций и переменные в этой программе в соответствии с лучшими практиками?

```
fun main() {  
    val Шаги = 4000  
    val caloriesBurned = PEDOMETERstepsTocalories(Steps);  
    println(«Ходьба на $Steps шагов сжигает $caloriesBurned калорий»)  
}  
  
fun PEDOMETERstepsTocalories(NumberOfStepS: Int): Double {  
    val CaloriesBURNEDforEachStep = 0.04  
    val TotalCALORIESburned = NumberOfStepS * CaloriesBURNEDforEachStep  
    return TotalCALORIESburned  
}
```

9. Сравните два числа

В современных мобильных телефонах есть встроенная функция, которая отслеживает экранное время, или время, которое вы проводите в телефоне каждый день.

В этом упражнении вы реализуете функцию, которая сравнивает время в минутах, проведенное вами на телефоне сегодня, со временем, проведенным вчера. Функция принимает два целочисленных параметра и возвращает булево значение.

Первый параметр содержит количество минут, потраченных вами сегодня, а второй - количество минут, потраченных вами вчера. Функция возвращает значение true, если вы провели больше времени на телефоне сегодня по сравнению со вчерашним днем. В противном случае она возвращает ложное значение.

Например, если вы вызвали функцию с такими именованными аргументами:

- timeSpentToday = 300 и timeSpentYesterday = 250, функция вернет истинное значение.
- timeSpentToday = 300 и timeSpentYesterday = 300, функция вернет ложное значение.
- timeSpentToday = 200 и timeSpentYesterday = 220, функция возвращает ложное значение.

Подсказка: оператор сравнения `>` возвращает истинное значение, если значение перед оператором больше значения после него. В противном случае он возвращает ложное значение.

10. Перенос дублирующегося кода в функцию

Эта программа отображает погоду для разных городов. Она включает в себя название города, высокую и низкую температуру в течение дня, а также вероятность дождя.

```
fun main() {  
    println(«Город: Анкара»)  
    println(«Низкая температура: 27, высокая температура: 31»)  
    println(«Вероятность дождя: 82%»)  
    println()  
  
    println(«Город: Токио»)  
    println(«Низкая температура: 32, высокая температура: 36»)  
    println(«Вероятность дождя: 10%»)  
    println()  
  
    println(«Город: Кейптаун»)  
    println(«Низкая температура: 59, высокая температура: 64»)  
    println(«Вероятность дождя: 2%»)  
    println()  
  
    println(«Город: Гватемала-Сити»)  
    println(«Низкая температура: 50, высокая температура: 55»)  
    println(«Вероятность дождя: 7%»)  
    println()  
}
```

В коде, выводящем погоду для каждого города, есть много общего. Например, есть фразы, которые повторяются несколько раз, такие как «Город:» и «Низкая температура:». Похожий, повторяющийся код создает риск ошибок в вашей программе. Для одного из городов вы можете допустить опечатку или забыть одну из деталей погоды.

Можете ли вы создать функцию, которая печатает данные о погоде для одного города, чтобы уменьшить количество повторений в функции `main()`, а затем сделать то же самое для остальных городов? Можете ли вы обновить функцию `main()`, чтобы вызвать функцию, которую вы создали для каждого города, и передать в качестве аргументов соответствующие данные о погоде?