Практическая работа: Основы Kotlin

Прежде чем начать

Теперь, когда вы приложили все усилия для изучения основ программирования на Kotlin, 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 как пользователь.

Необходимые условия

- Умение определять и вызывать функции.
- Знание основ программирования на Kotlin, включая переменные, функции println() и main().
- Знакомство с условными выражениями Kotlin, включая операторы и выражения if/else и when.
- Знакомство с лямбда-выражениями Kotlin
- Знание того, как работать с нулевыми переменными.
- Знание того, как создавать классы и объекты Kotlin.

2. Мобильные уведомления

Как правило, ваш телефон предоставляет вам сводку уведомлений.

В исходном коде, представленном в следующем фрагменте кода, напишите программу, которая печатает сводное сообщение, основанное на количестве полученных вами уведомлений. Сообщение должно содержать:

Точное количество уведомлений, если их меньше 100. 99+ - количество уведомлений, если их 100 или более.

```
fun main() {
    val morningNotification = 51
    val eveningNotification = 135

    printNotificationSummary(morningNotification)
    printNotificationSummary(eveningNotification)
}

fun printNotificationSummary(numberOfMessages: Int) {
    // Fill in the code.
}
```

Дополните функцию printNotificationSummary() так, чтобы программа выводила эти строки:

```
You have 51 notifications.
Your phone is blowing up! You have 99+ notifications.
```

3. Цена билета в кино

Цены на билеты в кино обычно различаются в зависимости от возраста кинозрителей.

На основе исходного кода, приведенного в следующем фрагменте кода, напишите программу, которая рассчитывает цены на билеты в зависимости от возраста:

Цена детского билета в 15 долларов для людей 12 лет и младше. Стандартный билет по цене $30 для людей в возрасте от 13 до 60 лет. По понедельникам для этой же возрастной группы снижайте цену стандартного билета до $25. Цена билета для пенсионеров составляет $20 для людей в возрасте 61 года и старше. Предположим, что максимальный возраст кинозрителя составляет 100 лет. Значение -1 указывает на то, что цена недействительна, если пользователь вводит возраст, выходящий за рамки возрастных спецификаций.

```
fun main() {
    val child = 5
    val adult = 28
    val senior = 87

    val isMonday = true

    println("The movie ticket price for a person aged $child is
\$${ticketPrice(child, isMonday)}.")
    println("The movie ticket price for a person aged $adult is
\$${ticketPrice(adult, isMonday)}.")
    println("The movie ticket price for a person aged $senior is
\$${ticketPrice(senior, isMonday)}.")
}

fun ticketPrice(age: Int, isMonday: Boolean): Int {
    // Fill in the code.
}
```

Дополните функцию ticketPrice() так, чтобы программа вывела эти строки:

```
The movie ticket price for a person aged 5 is $15.
The movie ticket price for a person aged 28 is $25.
The movie ticket price for a person aged 87 is $20.
```

4. Конвертер температуры

В мире существует три основные температурные шкалы: Цельсия, Фаренгейта и Кельвина.

На основе исходного кода, представленного в следующем фрагменте кода, напишите программу, которая преобразует температуру из одной шкалы в другую с помощью этих формул:

Цельсия в Фаренгейт: ° F = 9/5 (° C) + 32 Кельвин в Цельсий: ° C = K - 273,15 Перевод градуса Фаренгейта в градусы Кельвина: K = 5/9 (° F - 32) + 273,15 Обратите внимание, что метод String.format(«%.2f», /* измерение */) используется для преобразования числа в тип String с 2 десятичными знаками.

```
fun main() {
    // Fill in the code.
}


fun printFinalTemperature(
    initialMeasurement: Double,
    initialUnit: String,
    finalUnit: String,
    conversionFormula: (Double) -> Double
) {
    val finalMeasurement = String.format("%.2f",
conversionFormula(initialMeasurement)) // two decimal places
    println("$initialMeasurement degrees $initialUnit is $finalMeasurement degrees
$finalUnit.")
}
```

Дополните функцию main() так, чтобы она вызывала функцию printFinalTemperature() и выводила следующие строки. Вам нужно передать аргументы для температуры и формулы преобразования. Подсказка: возможно, вы захотите использовать значения Double, чтобы избежать усечения Integer при делении.

```
27.0 degrees Celsius is 80.60 degrees Fahrenheit.
350.0 degrees Kelvin is 76.85 degrees Celsius.
10.0 degrees Fahrenheit is 260.93 degrees Kelvin.
```

5. Каталог песен

Представьте, что вам нужно создать приложение для воспроизведения музыки.

Создайте класс, который может представлять структуру песни. Класс Song должен включать в себя следующие элементы кода:

Свойства для названия, исполнителя, года публикации и количества воспроизведений. Свойство, указывающее, является ли песня популярной. Если количество воспроизведений меньше 1 000,

считайте песню непопулярной. Метод, печатающий описание песни в таком формате: «[Название], исполненное [артист], было выпущено в [год публикации]».

6. Интернет-профиль

Часто от вас требуется заполнить профили на интернет-сайтах, которые содержат обязательные и необязательные поля. Например, вы можете добавить свою личную информацию и указать ссылку на других людей, которые направили вас для регистрации профиля.

На основе исходного кода, представленного в следующем фрагменте кода, напишите программу, которая распечатывает данные профиля человека .

```
fun main() {
    val amanda = Person("Amanda", 33, "play tennis", null)
    val atiqah = Person("Atiqah", 28, "climb", amanda)

    amanda.showProfile()
    atiqah.showProfile()
}


class Person(val name: String, val age: Int, val hobby: String?, val referrer:
Person?) {
    fun showProfile() {
       // Fill in code
    }
}
```

Завершите функцию showProfile() так, чтобы программа вывела эти строки:

```
Name: Amanda
Age: 33
Likes to play tennis. Doesn't have a referrer.

Name: Atiqah
Age: 28
Likes to climb. Has a referrer named Amanda, who likes to play tennis.
```

7. Складные телефоны

Обычно экран телефона включается и выключается при нажатии кнопки питания. В складном телефоне, напротив, основной внутренний экран не включается при нажатии кнопки питания.

В исходном коде, представленном в следующем фрагменте кода, напишите класс FoldablePhone, который наследуется от класса Phone. Он должен содержать следующее:

Свойство, указывающее, сложен ли телефон. Поведение функции switchOn(), отличное от поведения класса Phone, чтобы она включала экран только тогда, когда телефон не сложен. Методы для изменения состояния складывания.

```
class Phone(var isScreenLightOn: Boolean = false){
    fun switchOn() {
        isScreenLightOn = true
    }

    fun switchOff() {
        isScreenLightOn = false
    }

    fun checkPhoneScreenLight() {
        val phoneScreenLight = if (isScreenLightOn) "on" else "off"
        println("The phone screen's light is $phoneScreenLight.")
    }
}
```

8. Специальный аукцион

Обычно на аукционе цену на товар определяет участник, предложивший наибольшую цену. В этом специальном аукционе, если на товар нет претендента, он автоматически продается аукционному дому по минимальной цене.

В исходном коде, представленном в следующем фрагменте кода, вам дана функция auctionPrice(), принимающая в качестве аргумента нулевой тип Bid?

```
fun main() {
    val winningBid = Bid(5000, "Private Collector")

    println("Item A is sold at ${auctionPrice(winningBid, 2000)}.")
    println("Item B is sold at ${auctionPrice(null, 3000)}.")
}

class Bid(val amount: Int, val bidder: String)

fun auctionPrice(bid: Bid?, minimumPrice: Int): Int {
   // Fill in the code.
}
```

Дополните функцию auctionPrice() так, чтобы программа вывела эти строки:

```
Item A is sold at 5000.
Item B is sold at 3000.
```