№ 11 Kotlin. Типы. Функции. Операторы. Коллекции

Задание

1) Используя Converter to Kotlin **сконвертируйте** Java class и разберитесь с полученным текстом

```
public class StringDemo {
    public static void main(String[] args) {
        String palindrome = "Dot saw I was Tod";
        int len = palindrome.length();
        char[] tempCharArray = new char[len];
        char[] charArray = new char[len];

        for (int i = 0; i < len; i++) {
            tempCharArray[i] =
                palindrome.charAt(i);
        }

        for (int j = 0; j < len; j++) {
            charArray[j] =
                tempCharArray[len - 1 - j];
        }

        String reversePalindrome =
            new String(charArray);
        System.out.println(reversePalindrome);
    }
}
```

2) Определение переменных и констант
   a. Определите несколько переменных с val и var с явным указанием типа и без (Int, Double, String) на уровне файла.
   b. Выполните **преобразования переменных** из типа Byte в Int, из Int в String.
   c. Выведите их значения на консоль через строковый литерал с текстом и с ссылкой на переменные.
   d. Объявите константу.
   e. Объявите переменную **типа Int?** Введите с консоли число (или пустую строку).
3) Функции, функции расширения
   a. Напишите функцию sum **с переменным числом аргументов** типа Double, которая суммирует все переданные значения.
   b. Напишите функцию isValid проверки корректности ввода логина – пароля (параметры функции). Логин должно иметь формат email. Пароль от 6 до 12 символов, без пробелов. Логин-пароль не могут быть пустыми. Для этой

проверки напишите **локальную функцию** notNull.Функция notNull должна иметь **тело-выражение** (без блока) на основе if.

c. Задайте через **перечисление праздничные** дни в году. Напишите функцию с **использованием *when*** для проверки по введенной дате (день, месяц, год) - будний или праздничный день. Предусмотрите вариант, когда пользователь передал null строку или формат не соответствует.

d. Допишите функцию

fun doOperation (a:Int , b:Int, operation:Char): Double

Используя конструкцию when, вычислите значение операции, указанной в operation. Функция должна корректно обрабатывать любую допустимую в Kotlin бинарную операцию. Если операция допустима – генерируйте исключение. Продемонстрируйте работу функции.

e. Реализуйте функцию indexOfMax (), чтобы она возвращала индекс самого большого элемента в массиве, или null, если массив пуст или таких элементов несколько. Сделайте ее потом **функцией расширения** для IntArray

```
fun indexOfMax(a: IntArray): Int? {
}
```

f. Напишите **функцию расширения** coincidence для String, которая проверяет сколько позиций совпало со строкой перегаданной в аргументе и возвращает количество совпавших символов.

4) Циклы и диапазоны

a. Определите функцию вычисления факториала в двух вариантах: 1) **с циклом и диапазоном**

fun factorial(n: Int): Double

2) рекурсивную на основе индуктивного определения факториала n! = n(n-1)!

fun factorial(n: Int): Double =

b. Определите функцию isPrime, проверки является ли число простым. (Напишите ее оптимально. В частности, достаточно проверить делимость числа n на все числа в интервале от 2 до n/2, так как на большие числа n всё равно делится не будет. Достаточно ограничится интервалом от 2 до √n — если n и делится на какое-то большее √n число (например, 50 делится на 10), то оно будет делится и на какое-то меньшее число (в данном случае, 50 делится на 5=50/10).) С помощью isPrime узнайте, сколько существует простых чисел, меньших 10 000. Первые 20 простых чисел поместите в список, следующие 10 в массив.

5) Коллекции и лямбды

a. Напишите **2 лямбды** для передачи в функцию, чтобы проверить, содержит ли задуманное коллекция. Функция ***any*** получает предикат в качестве аргумента и возвращает true, если хотя бы один элемент удовлетворяет предикату.

```
fun containsIn(collection: Collection<Int>): Boolean = collection.any { TODO() }
```

b. Используя **listOf** сформируйте список целых. Примените несколько разных способов добавления элемента в коллекцию (add, +=). Оставьте только уникальные элементы. Отфильтруйте и оставьте только нечетные. Выведите элементы через **forEach**. Передайте **ссылку на функцию** проверки на простое число в **filter** для проверки элементов списка.

```
val numbers = listOf(1, 2, 3,…..)
 println(numbers.filter(::isPrime))
```

Примените к списку **find, groupBy, all, any**
Выполните **деструктуризацию** первых 2-х элементов списка.

c. Сформируйте map с фамилией и количеством правильных ответов в тесте (число от 1 до 40 максимально). Выполните преобразование – замените число правильных ответов на оценку. Используйте следующие критерии
40 - 10;
39 - 9 ;
38 – 8;
37..35 - 7;

34..32 - 6;
31..29 - 5;
28..25 - 4;
24..22 - 3;
21..19 - 2;
18..0 - 1.

Подсчитайте количество всех оценок по категориям, проверьте есть ли неуды.

Вопросы при защите:

1. Назвоните основные преимущества языка Kotlin.
2. Куда может компилироваться Kotlin?
3. Что такое функциональное программирование?
4. Расскажите как определить функцию. Должна ли она принадлежать классу?
5. Приведите пример определения функции путем присвоения выражения (тело-выражение), например if, try …
6. Как объявить переменную? Нужен ли тип? В чем отличие val и var?
7. Можно ли поменять объект, на который указывает val, может быть изменяемым?
8. Приведите пример использования конструкции when.
9. Как используются is и as?
10. Какие циклы можно использовать в Kotlin?
11. Приведите пример задания диапазона. Какие функции можно использовать в интервалах?
12. Как происходит обработка исключений?
13. Как определить тип способный хранить null?
14. Как и в каких случаях можно использовать операторы ?: !! ?.
15. Различает ли Kotlin примитивные типы и типы-оберток? Как происходит компиляция типов с поддержкой null?
16. Выполняет ли Kotlin автоматическое преобразования чисел из одного типа в другой?
17. Расскажите про типы Any и Any? .
18. Расскажите про тип Unit и Nothing.
19. Как компилируются функции верхнего уровня? Создается ли для них класс?
20. Какова роль функций расширений в Kotlin? Приведите пример такой функции.
21. Как определяется свойство-расширения?

22. Как задать функцию с переменным числом параметров?
23. Для чего используются локальный (вложенные) функции? Как их вызвать?
24. Задайте функциональный тип?
25. Как используется ключевое слово it в анонимной функции?
26. Как задать и использовать ссылку на функцию?
27. Приведите пример использования функций apply, let, run, with, also, takeIf.
28. Как в Kotlin записываются лямбда – выражения?
29. Как создать изменяемую и неизменяемую коллекцию List, Set, Map?
30. Как добавить, удалить элементы из коллекций, преобразовать к другому типу?
31. Приведите пример деструктуризации списка.
32. Как создать массив значений примитивного типа без оберток?