

# **Лабораторная работа №8**

## **Наследование классов**

### **1 Цель работы**

1.1 Изучить процесс разработки дочерних классов в приложениях на Kotlin.

### **2 Литература**

2.1 Kotlin | Руководство. metanit.com – Текст : электронный //metanit.com, 2023. – URL: <https://metanit.com/kotlin/tutorial/> – гл.4.

### **3 Подготовка к работе**

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

### **4 Основное оборудование**

4.1 Персональный компьютер.

### **5 Задание**

Все созданные классы, их функции и свойства необходимо протестировать.

5.1 Создание родительского и дочернего классов

5.1.1 Создать родительский класс Человек с:

- свойствами имя и возраст,

- конструктором для присвоения значений свойствам.

5.1.2 Создать дочерний класс Студент с:

- свойством группа

- конструктором для присвоения значений всем свойствам (с использованием родительского конструктора)

5.2 Переопределение методов и свойств. Во всех методах и свойствах строки вывода должны отличаться (например, комментариями в строке)

5.2.1 Переопределить в родительском и дочернем классах метод toString(). В дочернем переопределении использовать реализацию из родительского класса.

5.2.2 Создать в родительском классе свойство на чтение для возвращения в виде строки всей информации о человеке. Реализовать его переопределение в дочернем классе для возврата информации обо всех полях.

5.2.3 Создать в родительском классе метод для вывода всей информации о человеке. Реализовать его переопределение в дочернем классе для вывода информации обо всех свойствах. Значение каждого свойства должно быть выведено с новой строки. (Необходимо в дочерней реализации вызывать родительскую)

5.3 Создать абстрактный класс Фигура, в котором объявить следующие элементы:

- открытый абстрактный метод для возврата площади фигуры;

- открытый абстрактный метод для возврата периметра фигуры;
- открытый абстрактный метод для вывода информации о фигуре;
- открытое абстрактное свойство, возвращающее название фигуры.

Создать дочерний класс, в котором реализовать абстрактные методы и свойство и добавить в класс свойства и конструкторы. Задание выполняется по вариантам:

- прямоугольник (для нечетного варианта);
- круг (для четного варианта).

#### 5.4 Создание интерфейсов

5.4.1 Создать обобщенный интерфейс `DataSource`, предоставляющий метод `getNext()` возвращающий следующий элемент из некоторого источника данных.

5.4.2 Создать класс `RandomDataSource`, реализующий интерфейс `DataSource`, и возвращающий случайные элементы

5.4.3 Создать класс `StaticDataSource`, реализующий интерфейс `DataSource`, и возвращающий элементы некоторого массива, передаваемого в конструкторе.

5.4.4 Протестировать классы создав функцию для вывода 10 элементов принимающую объект `DataSource` и вызвав ее.

#### 5.5 Делегирование интерфейсов

5.5.1 Создать интерфейс Сервиса доставки с полями имени сервиса и методом `deliver`, принимающим номер заказа

5.5.2 Реализовать интерфейс в классе `FoodService`, представляющем сервис доставки еды. В методе `deliver` выводить на консоль "Заказ ... передан в доставку через ..."

5.5.3 Реализовать класс Приложения доставки и делегировать в нем интерфейс доставки. Добавить в класс метод для создания заказа, вызывающий метод `deliver`.

### 6 Порядок выполнения работы

- 6.1 Выполнить все задания из п.5.
- 6.2 Ответить на контрольные вопросы.

### 7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

### 8 Контрольные вопросы

- 8.1 Что такое «наследование»?
- 8.2 Сколько родительских классов может быть у класса в Kotlin?
- 8.3 Сколько интерфейсов может реализовывать класс в Kotlin?
- 8.4 Как указать родительский класс на Kotlin?
- 8.5 Как переопределить реализацию метода в дочернем классе на Kotlin?

8.6 Как вызвать реализацию родительского метода в дочернем классе на Kotlin?

8.7 Как указать, что класс может быть родительским?

8.8 Как выполнить делегирование интерфейса?