

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

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.

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 интерфейс IPrinter, в котором объявить:

- метод void Print().

5.4.2 Создать интерфейс IFigure, в котором объявить следующие элементы:

- метод для возврата площади фигуры;
- метод для возврата периметра фигуры;
- метод для вывода информации о фигуре;
- свойство, возвращающее название фигуры.

5.5 Реализация интерфейса

5.5.1 Реализовать интерфейс IPrinter в классе из таблицы 2 (по вариантам, вариант – номер компьютера). Метод должен выводить на консоль значения всех свойств класса.

5.5.2 Реализация интерфейса IFigure. Реализовать интерфейс IFigure в классе из таблицы 2. Для нахождения периметра и площади использовать только указанные поля класса. Свойство должно возвращать название фигуры, которое указано в таблице.

5.5.3 Протестировать работу методов и свойства, создав объекты класса и вызвав у них реализованные методы и свойство.

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 указать, что у класса нельзя создавать дочерние классы?

9 Приложение

Таблица 2

№	Наименование	Свойства
1,8	Параллелограмм	Основание, Высота, Боковая сторона
2,9	Трапеция	Основание А, Основание Б, Высота, Боковая сторона В, Боковая сторона Д
3,10	Круг	Радиус
4,11	Квадрат	Сторона
5,12	Треугольник	Сторона А, Сторона Б, Сторона В, Высота к стороне А
6,13	Прямоугольник	Сторона А, Сторона Б
7,14	Эллипс	Большая полуось, Малая полуось