

Программирование на Java

Практическая работа №4 (part 1)

Представление объектов и классов. (6,7 неделя)

Задание 1: Создание класса «Книга» и работа с объектами класса "Книга"

Описание: Создайте класс Book, который будет представлять книгу. Класс должен содержать следующие поля:

- title (название книги)
- author (автор книги)
- price (цена книги)

Создайте несколько объектов класса Book и заполните их данными.

Условия:

Реализуйте конструктор для инициализации полей класса.

Добавьте методы для получения значений полей (геттеры) и метод для вывода информации о книге.

Создайте не менее трех объектов книги.

Выведите информацию о каждой книге с помощью метода, который вы реализовали в классе Book.

```
public class Main {  
    public static void main(String[] args) {  
        Book book1 = new Book("1984", "Джордж Оруэлл", 299.9);  
        Book book2 = new Book("Война и мир", "Лев Толстой", 499.5);  
        Book book3 = new Book("Мастер и Маргарита", "Михаил Булгаков", 350);  
  
        book1.displayInfo();  
        book2.displayInfo();  
        book3.displayInfo();  
    }  
}
```

Рисунок 1 – Пример главного класса задания «Работа с объектами класса Книга»

```
Название: 1984, Автор: Джордж Оруэлл, Цена: 299.99  
Название: Война и мир, Автор: Лев Толстой, Цена: 499.5  
Название: Мастер и Маргарита, Автор: Михаил Булгаков, Цена: 350.0
```

Рисунок 2 – Пример выполнения программы «Работа с объектами класса Книга»

Задание 2: Сравнение «Студент»

Описание: Создайте класс Student, который будет представлять студента. Класс должен содержать следующие поля:

- name (имя студента)
- age (возраст)
- major (специальность)

Создайте массив объектов класса Student, который позволяет сравнивать студентов по возрасту.

Условия:

Реализуйте конструктор, геттеры и сеттеры для всех полей.

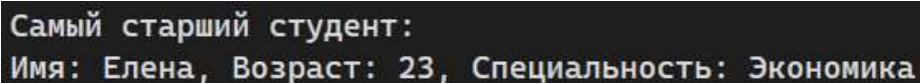
Добавьте метод для вывода информации о студенте.

Реализуйте метод, который будет находить самого старшего студента в массиве.

Выведите информацию о самом старшем студенте.

```
public class Main {  
    public static void main(String[] args) {  
        Student[] students = {  
            new Student("Алексей", 20, "Информатика"),  
            new Student("Мария", 22, "Филология"),  
            new Student("Иван", 21, "Математика"),  
            new Student("Елена", 23, "Экономика")  
        };  
  
        Student oldestStudent = Student.findOldestStudent(students);  
        System.out.println("Самый старший студент:");  
        oldestStudent.displayInfo();  
    }  
}
```

Рисунок 3 – Пример главного класса задания «Сравнение студентов»



Самый старший студент:
Имя: Елена, Возраст: 23, Специальность: Экономика

Рисунок 4 – Пример выполнения задания «Сравнение студентов»

Задание 3: Класс «Продукт» с методами

Описание: Создайте класс Product, который будет представлять продукт в магазине. Класс должен содержать следующие поля:

- name (название продукта)
- quantity (количество на складе)
- price (цена)

Условия:

Реализуйте методы для:

- увеличения или уменьшения количества товара;
- изменения цены;
- вывода информации о продукте;

```

public class Main {
    public static void main(String[] args) {
        Product product = new Product("Яблоко", 100, 0.5);

        product.displayInfo();
        product.increaseQuantity(20);
        product.decreaseQuantity(30);
        product.changePrice(0.7);
        product.displayInfo();
    }
}

```

Рисунок 5 – Пример главного класса задания «Класс Продукт с методами»

```

Продукт: Яблоко
Количество на складе: 100
Цена: 0.5
Количество товара увеличено на 20. Новое количество: 120
Количество товара уменьшено на 30. Новое количество: 90
Цена изменена на 0.7
Продукт: Яблоко
Количество на складе: 90
Цена: 0.7

```

Рисунок 6 – Пример выполнения задания «Класс Продукт с методами»

Задание 4: Наследование

Описание: Создайте класс `ElectronicDevice`, а затем создайте подкласс `Smartphone`, который наследует от `ElectronicDevice`.

Условия:

Класс `ElectronicDevice` должен содержать поля:

- `brand` (бренд)
- `model` (модель)
- `price` (цена)

Класс `Smartphone` должен добавлять поле:

- `operatingSystem` (операционная система)

Реализуйте методы для вывода информации о каждом устройстве.

```

class Main {
    public static void main(String[] args) {
        ElectronicDevice device = new ElectronicDevice("Samsung", "Galaxy Tab", 299.99);
        Smartphone smartphone = new Smartphone("Apple", "iPhone 14", 999.99, "iOS");

        System.out.println("Информация об устройстве:");
        device.displayInfo();

        System.out.println("\nИнформация о смартфоне:");
        smartphone.displayInfo();
    }
}

```

Рисунок 7 – Пример главного класса задания «Наследование»

```
Информация об устройстве:  
Бренд: Samsung  
Модель: Galaxy Tab  
Цена: 299.99  
  
Информация о смартфоне:  
Бренд: Apple  
Модель: iPhone 14  
Цена: 999.99  
Операционная система: iOS
```

Рисунок 8 – Пример выполнения задания «Наследование»

Задание 5: Полиморфизм

Описание: Создайте класс `Animal` с методом `makeSound()`, а затем создайте подклассы `Dog` и `Cat`, переопределяющие этот метод.

Условия:

Напишите программу, которая создает массив объектов `Animal`, заполняет его экземплярами `Dog` и `Cat` и вызывает метод `makeSound()` для каждого объекта.

```
public class Main {  
    public static void main(String[] args) {  
        Animal[] animals = new Animal[3];  
  
        animals[0] = new Dog();  
        animals[1] = new Cat();  
        animals[2] = new Dog();  
  
        for (Animal animal : animals) {  
            animal.makeSound();  
        }  
    }  
}
```

Рисунок 9 – Пример главного класса задания «Полиморфизм»

```
Гав!  
Мяу!  
Гав!
```

Рисунок 10 – Пример выполнения задания «Полиморфизм»

Ресурсы

1. Принципы ООП Java: <https://javarush.com/groups/posts/principy-oop>
2. Основные принципы ООП: <https://training.epam.kz/ru/blog/275>
3. ООП в Java: четыре принципа с примерами: <https://highload.today/oop-v-java-chetyre-printsipa-s-primerami/>