Ускова_Екатерина

Задание 1:

Создать класс, который содержит 3 параметра:

- целочисленное значение - количество товаров
- вещественное значение - цена товара
- вещественное значение - скидка на товар

Сам класс должен иметь описание с многострочным комментарием. Каждое поле должно содержать однострочный комментарий с коротким описанием.

Сделать метод, который принимает объект на вход и выполняет подсчеты: вывести на экран общую сумму покупки без скидки и сумму со скидкой.

Создать второй класс, который будет содержать метод main. Сделать 3 вызова метода с различными объектами (количество товаров и сумма должны быть больше нуля) и скидками:

1) 0,75%
2) 42,575%
3) 59,1%

Итоговая сумма должна быть округлена до 2 знаков после запятой.

Ответ:

```
public class Product {
    int quantity;      //количество товаров
    double price;      //цена товара
    double discount;   //скидка на товар в процентах

    //конструктор
    public Product(int quantity, double price, double discount) {
        this.quantity = quantity;
        this.price = price;
        this.discount = discount;
    }

    //метод для подсчета общей суммы без скидки и со скидкой
    public void calculateTotal() {
        double totalWithoutDiscount = quantity * price;
        double totalWithDiscount = totalWithoutDiscount * (1 - discount / 100);
      //вывод округленных значений

        System.out.println("Без скидки: " + String.format("%.2f",
totalWithoutDiscount));
        System.out.println("Со скидкой: " + String.format("%.2f",
totalWithDiscount));
    }
}
```

Ускова_Екатерина

```
    public class Main {

    public static void main(String[] args) {
        //создание объектов
        Product product1 = new Product(10, 100, 0.75);
        Product product2 = new Product(5, 200, 42.575);
        Product product3 = new Product(3, 300, 59.1);

        /вызов метода для каждого товара
        System.out.println("Товар 1:");
        product1.calculateTotal();

        System.out.println("\nТовар 2:");
        product2.calculateTotal();

        System.out.println("\nТовар 3:");
        product3.calculateTotal();
    }
}
```

Вывод:

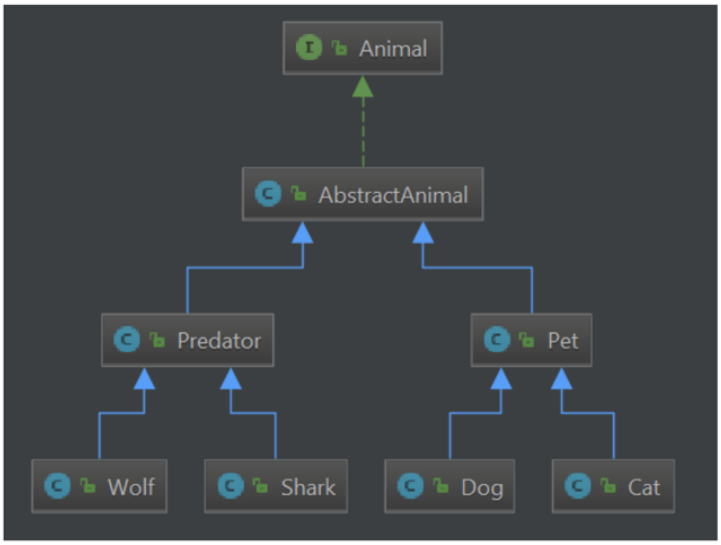
```
PS C:\Users\yneiv\Desktop\стажировки\MTS QA\homework>
able-preview' '-XX:+ShowCodeDetailsInExceptionMessages'
Товар 1:
Без скидки: 1000,00
Со скидкой: 992,50

Товар 2:
Без скидки: 1000,00
Со скидкой: 574,25

Товар 3:
Без скидки: 900,00
Со скидкой: 368,10
PS C:\Users\yneiv\Desktop\стажировки\MTS QA\homework>
```

Ускова_Екатерина

Задание 2:

Построить иерархию животных в виде наследуемых классов.

Пример:



Ответ:

```java
public abstract class AbstractAnimal implements Animal {
    String name; // Имя животного

    public AbstractAnimal(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }
}
```

```java
public interface Animal {
    void makeSound(); // Метод для издания звука
}
```

```java
public class Predator extends AbstractAnimal {
    public Predator(String name) {
        super(name);
    }

    @Override
    public void makeSound() {
        System.out.println(getName() + " издаёт угрожающие звуки.");
    }
}
```

```java
public class Pet extends AbstractAnimal {
    public Pet(String name) {
        super(name);
    }

    @Override
    public void makeSound() {
        System.out.println(getName() + " делает дружелюбные звуки/жесты.");

    }
}
```

```java
public class Wolf extends Predator {
    public Wolf(String name) {
        super(name);
    }

    @Override
    public void makeSound() {
        System.out.println(getName() + " скалит зубы");

    }
}
```

```java
public class Shark extends Predator {
    public Shark(String name) {
        super(name);
    }

    @Override
    public void makeSound() {
        System.out.println(getName() + " молчалива, но зубы остренькие");

    }
}
```

```java
public class Dog extends Pet {
    public Dog(String name) {
        super(name);
    }

    @Override
    public void makeSound() {
        System.out.println(getName() + " веляет хвостом");

    }
}
```

```
ublic class Cat extends Pet {
    public Cat(String name) {
        super(name);
    }

    @Override
    public void makeSound() {
        System.out.println(getName() + " мурлычит");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        // Создаём объекты животных
        Animal wolf = new Wolf("Волк");
        Animal shark = new Shark("Акула");
        Animal dog = new Dog("Собака");
        Animal cat = new Cat("Кошка");

        // Вызываем метод makeSound() для каждого животного
        wolf.makeSound();
        shark.makeSound();
        dog.makeSound();
        cat.makeSound();
    }
}
```

Вывод:

```
PS C:\Users\yneiv\Desktop\стажировки\MTS QA\homework>  c:
able-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '
Волк скалит зубы
Акула молчалива, но зубы остренькие
Собака веляет хвостом
Кошка мурлычит
PS C:\Users\yneiv\Desktop\стажировки\MTS QA\homework>
```