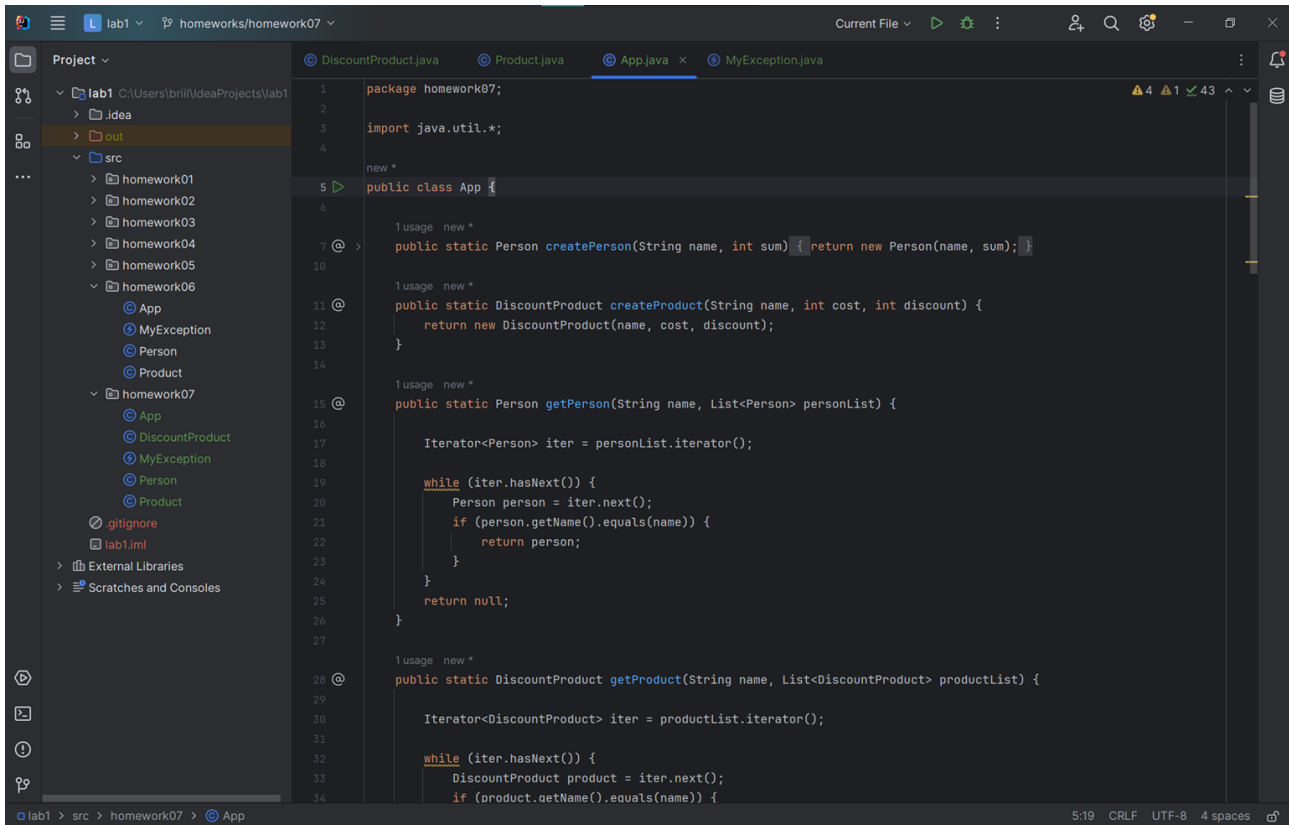
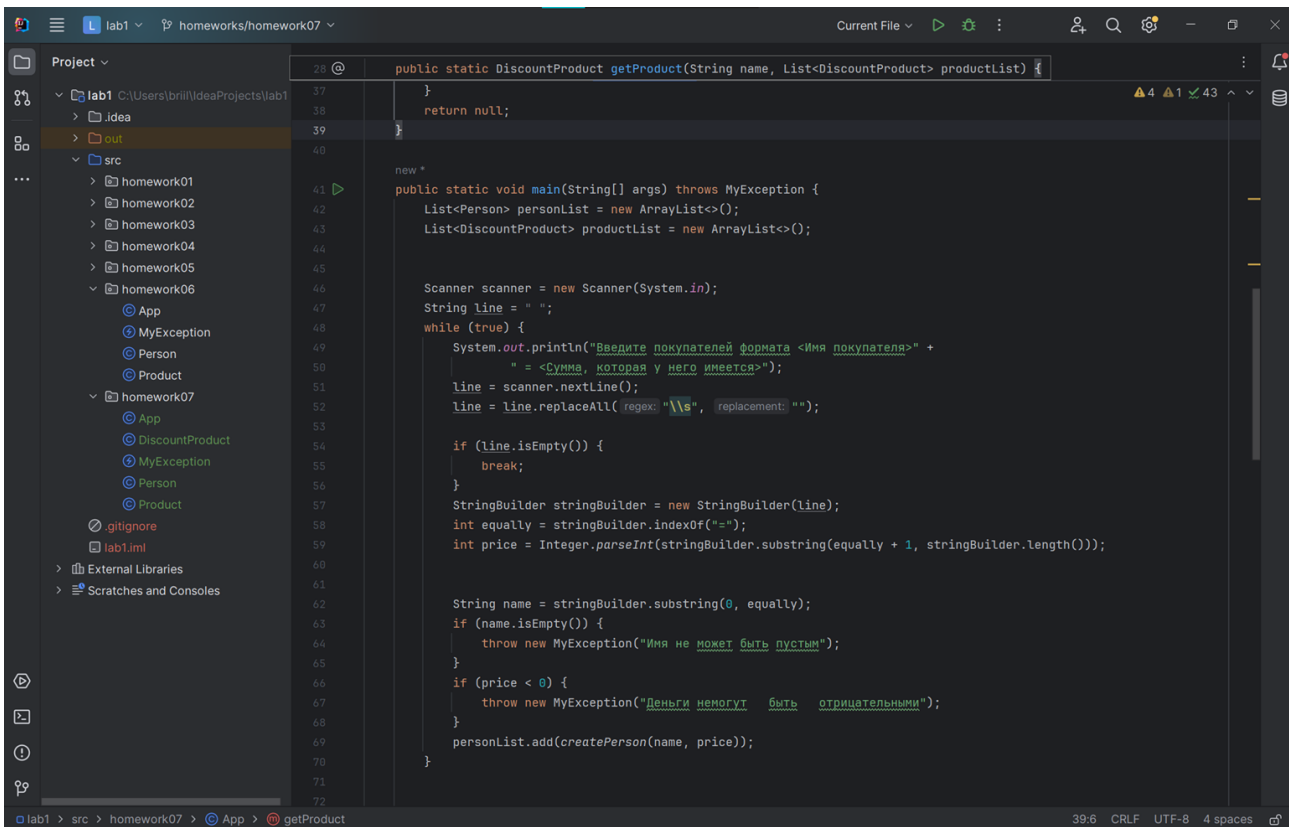


Класс App, демонстрирующий как работает класс DiscountProduct

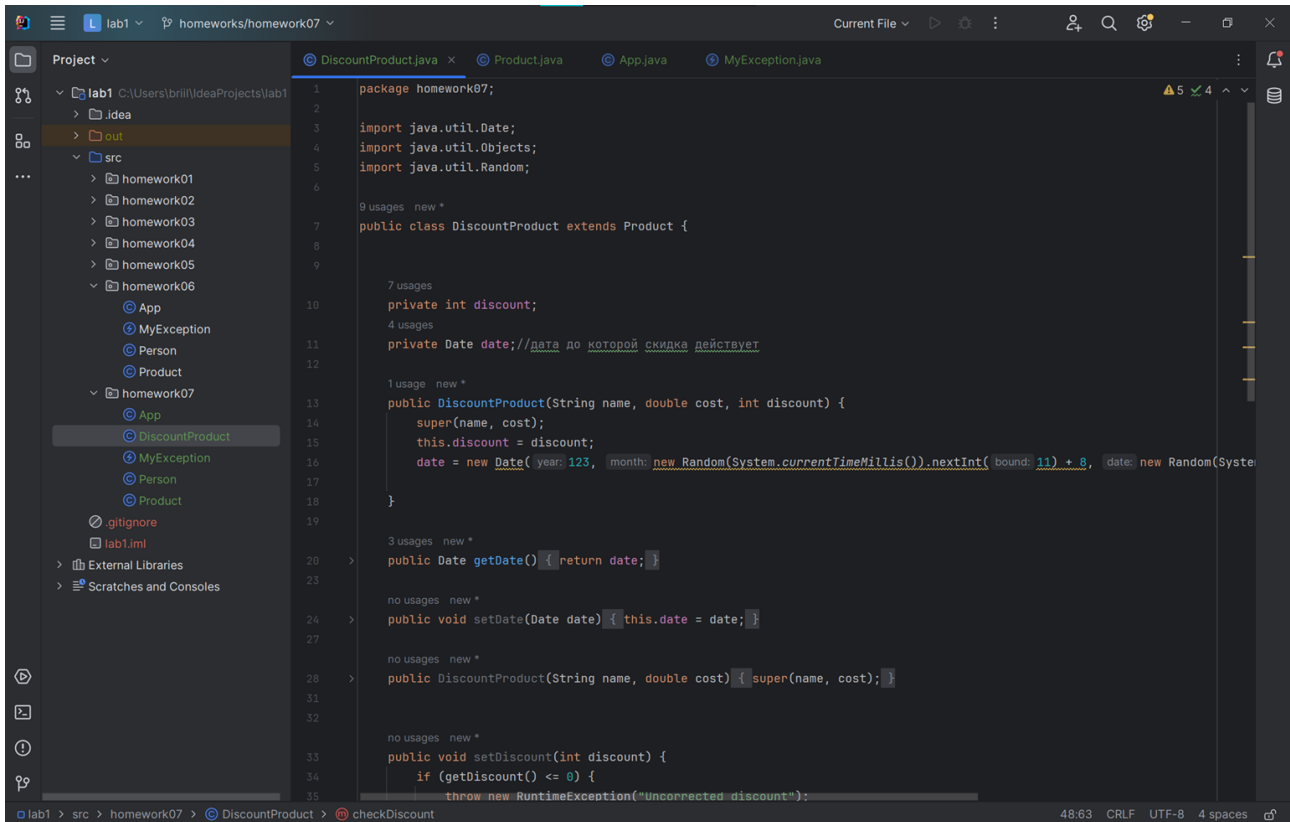


```
1 package homework07;
2
3 import java.util.*;
4
5 new *
6 public class App {
7
8     1 usage new *
9     public static Person createPerson(String name, int sum) { return new Person(name, sum); }
10
11     1 usage new *
12     public static DiscountProduct createProduct(String name, int cost, int discount) {
13         return new DiscountProduct(name, cost, discount);
14     }
15
16     1 usage new *
17     public static Person getPerson(String name, List<Person> personList) {
18
19         Iterator<Person> iter = personList.iterator();
20
21         while (iter.hasNext()) {
22             Person person = iter.next();
23             if (person.getName().equals(name)) {
24                 return person;
25             }
26         }
27         return null;
28     }
29
30     1 usage new *
31     public static DiscountProduct getProduct(String name, List<DiscountProduct> productList) {
32
33         Iterator<DiscountProduct> iter = productList.iterator();
34
35         while (iter.hasNext()) {
36             DiscountProduct product = iter.next();
37             if (product.getName().equals(name)) {
```

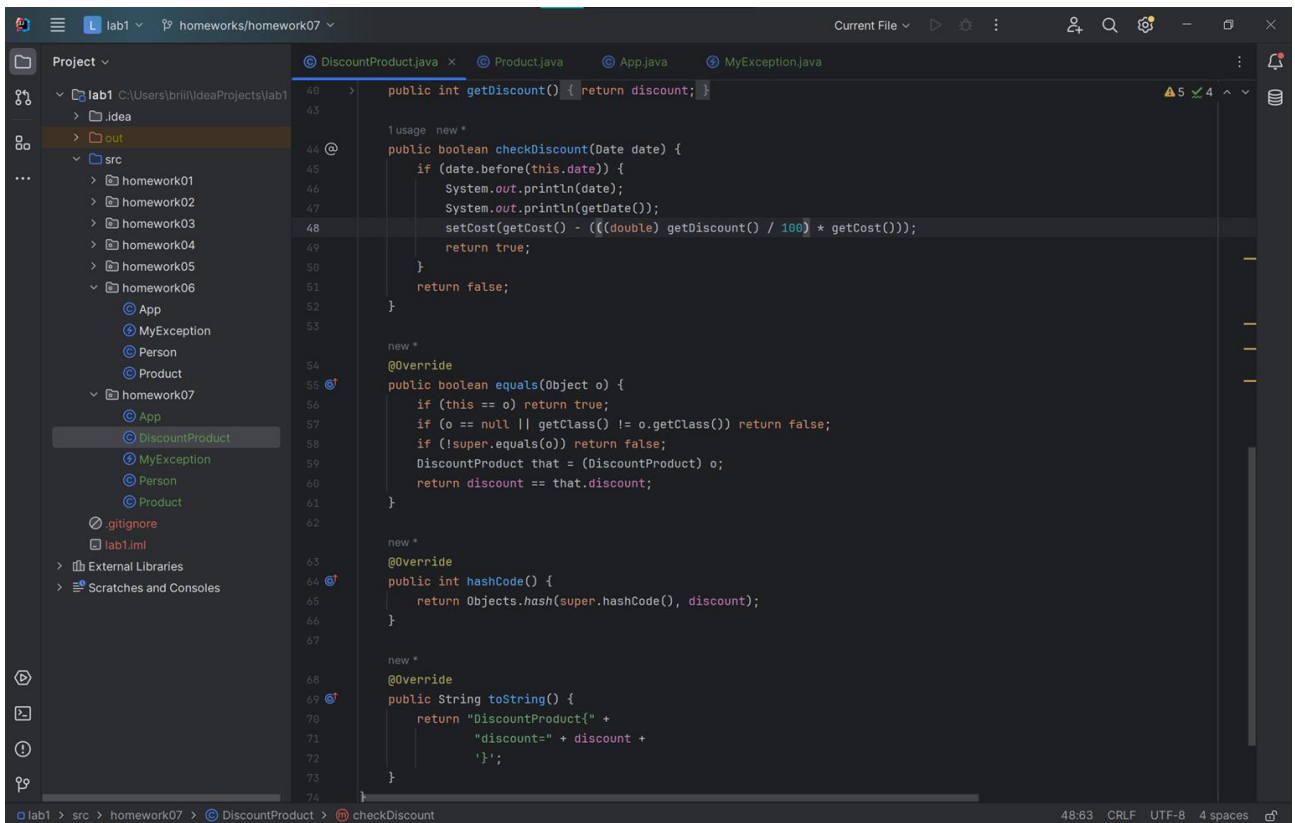


```
28 @
29 public static DiscountProduct getProduct(String name, List<DiscountProduct> productList) {
30
31     }
32     return null;
33 }
34
35 new *
36 public static void main(String[] args) throws MyException {
37     List<Person> personList = new ArrayList<>();
38     List<DiscountProduct> productList = new ArrayList<>();
39
40     Scanner scanner = new Scanner(System.in);
41     String line = " ";
42     while (true) {
43         System.out.println("Введите покупателя формата <Имя покупателя> +
44             " = <Сумма, которая у него имеется>");
45         line = scanner.nextLine();
46         line = line.replaceAll(regex: "\\s", replacement: "");
47
48         if (line.isEmpty()) {
49             break;
50         }
51         StringBuilder stringBuilder = new StringBuilder(line);
52         int equally = stringBuilder.indexOf("=");
53         int price = Integer.parseInt(stringBuilder.substring(equally + 1, stringBuilder.length()));
54
55         String name = stringBuilder.substring(0, equally);
56         if (name.isEmpty()) {
57             throw new MyException("Имя не может быть пустым");
58         }
59         if (price < 0) {
60             throw new MyException("Деньги не могут быть отрицательными");
61         }
62         personList.add(createPerson(name, price));
63     }
64 }
```

Класс DiscountProduct

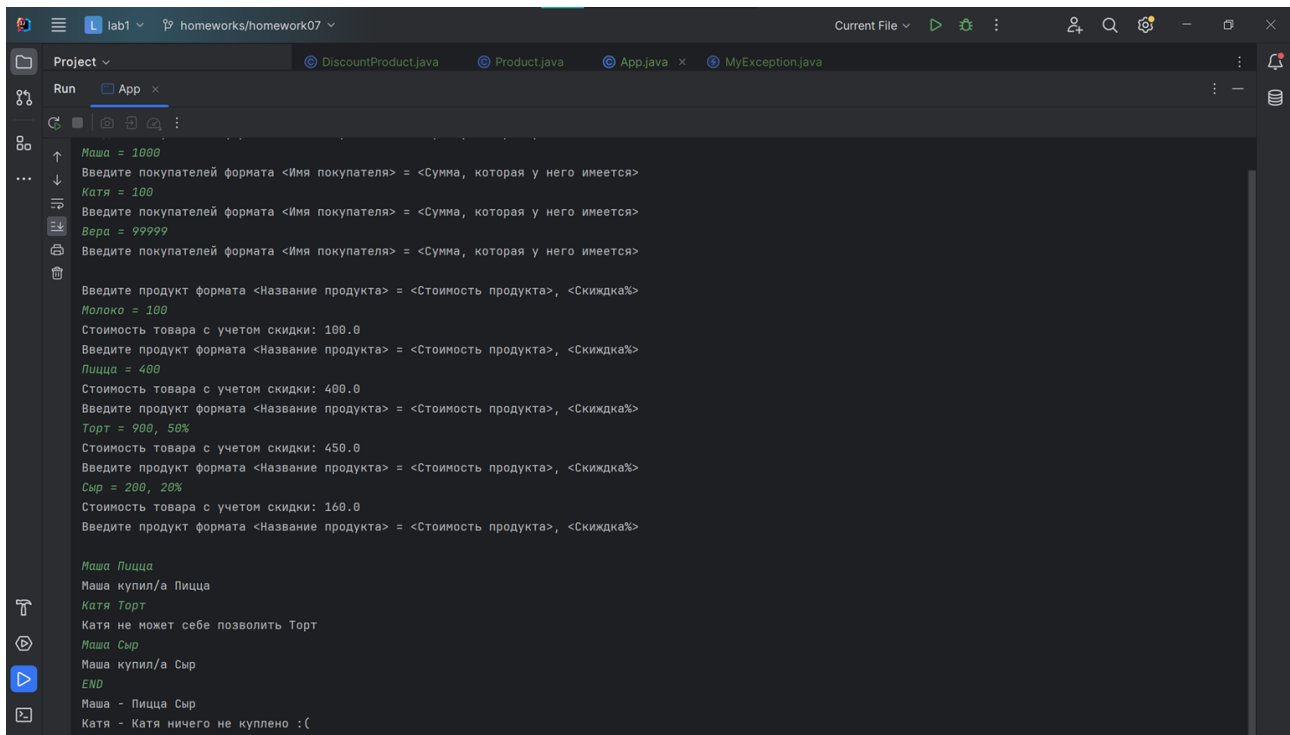


```
1 package homework07;
2
3 import java.util.Date;
4 import java.util.Objects;
5 import java.util.Random;
6
7
8
9
10 public class DiscountProduct extends Product {
11
12     7 usages new *
13     private int discount;
14     4 usages
15     private Date date; // дата до которой скидка действует
16
17     1 usage new *
18     public DiscountProduct(String name, double cost, int discount) {
19         super(name, cost);
20         this.discount = discount;
21         date = new Date( year: 123, month: new Random(System.currentTimeMillis()).nextInt( bound: 11) + 8, date: new Random(System.currentTimeMillis()).nextInt( bound: 28) + 1);
22     }
23
24     3 usages new *
25     public Date getDate() { return date; }
26
27     no usages new *
28     public void setDate(Date date) { this.date = date; }
29
30     no usages new *
31     public DiscountProduct(String name, double cost) { super(name, cost); }
32
33     no usages new *
34     public void setDiscount(int discount) {
35         if (getDiscount() <= 0) {
36             throw new RuntimeException("Unconnected discount");
37         }
38     }
39 }
```



```
40 public int getDiscount() { return discount; }
41
42
43
44 1 usage new *
45 public boolean checkDiscount(Date date) {
46     if (date.before(this.date)) {
47         System.out.println(date);
48         System.out.println(getDate());
49         setCost(getCost() - (((double) getDiscount() / 100) * getCost()));
50         return true;
51     }
52     return false;
53 }
54
55 new *
56 @Override
57 public boolean equals(Object o) {
58     if (this == o) return true;
59     if (o == null || getClass() != o.getClass()) return false;
60     if (!super.equals(o)) return false;
61     DiscountProduct that = (DiscountProduct) o;
62     return discount == that.discount;
63 }
64
65 new *
66 @Override
67 public int hashCode() {
68     return Objects.hash(super.hashCode(), discount);
69 }
70
71 new *
72 @Override
73 public String toString() {
74     return "DiscountProduct{" +
75         "discount=" + discount +
76         '}';
77 }
78 }
```

Демонстрация работы в консоли



```
Project ▾ DiscountProduct.java Product.java App.java × MyException.java
Run ▾ App ×
Masha = 1000
Введите покупателей формата <Имя покупателя> = <Сумма, которая у него имеется>
Katya = 100
Введите покупателей формата <Имя покупателя> = <Сумма, которая у него имеется>
Vera = 99999
Введите покупателей формата <Имя покупателя> = <Сумма, которая у него имеется>

Введите продукт формата <Название продукта> = <Стоимость продукта>, <Скидка%>
Молоко = 100
Стоимость товара с учетом скидки: 100.0
Введите продукт формата <Название продукта> = <Стоимость продукта>, <Скидка%>
Пицца = 400
Стоимость товара с учетом скидки: 400.0
Введите продукт формата <Название продукта> = <Стоимость продукта>, <Скидка%>
Торт = 900, 50%
Стоимость товара с учетом скидки: 450.0
Введите продукт формата <Название продукта> = <Стоимость продукта>, <Скидка%>
Сыр = 200, 20%
Стоимость товара с учетом скидки: 160.0
Введите продукт формата <Название продукта> = <Стоимость продукта>, <Скидка%>

Masha Пицца
Masha купил/а Пицца
Katya Торт
Katya не может себе позволить Торт
Masha Сыр
Masha купил/а Сыр
END
Masha - Пицца Сыр
Katya - Катя ничего не куплено :(
```