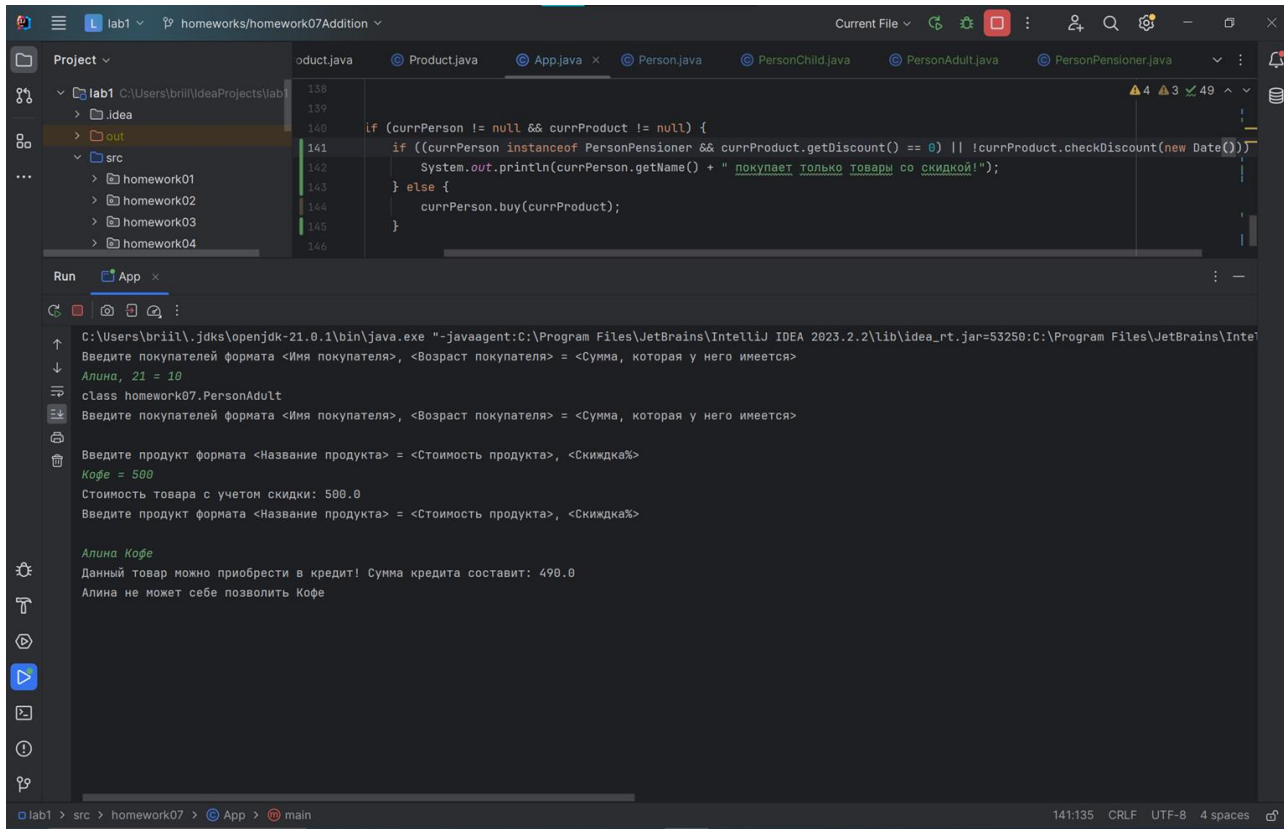
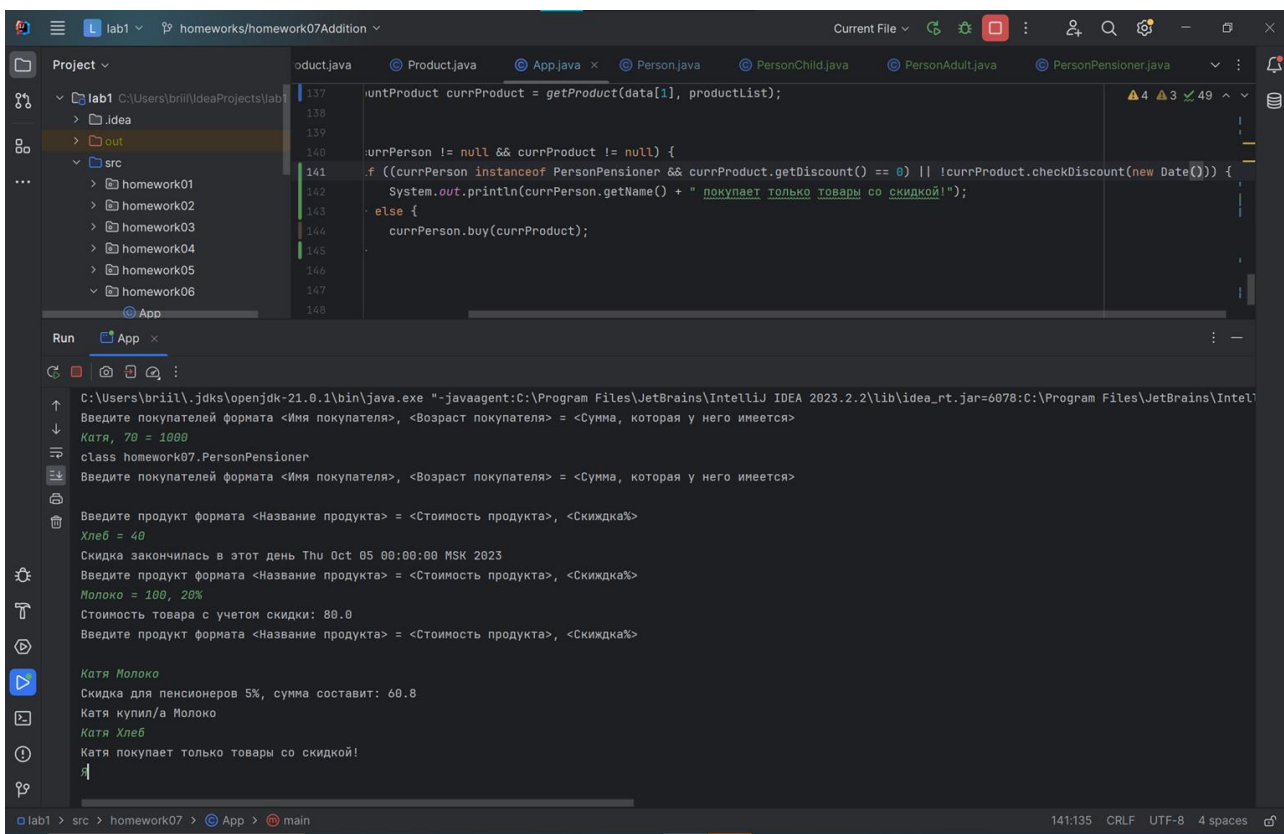
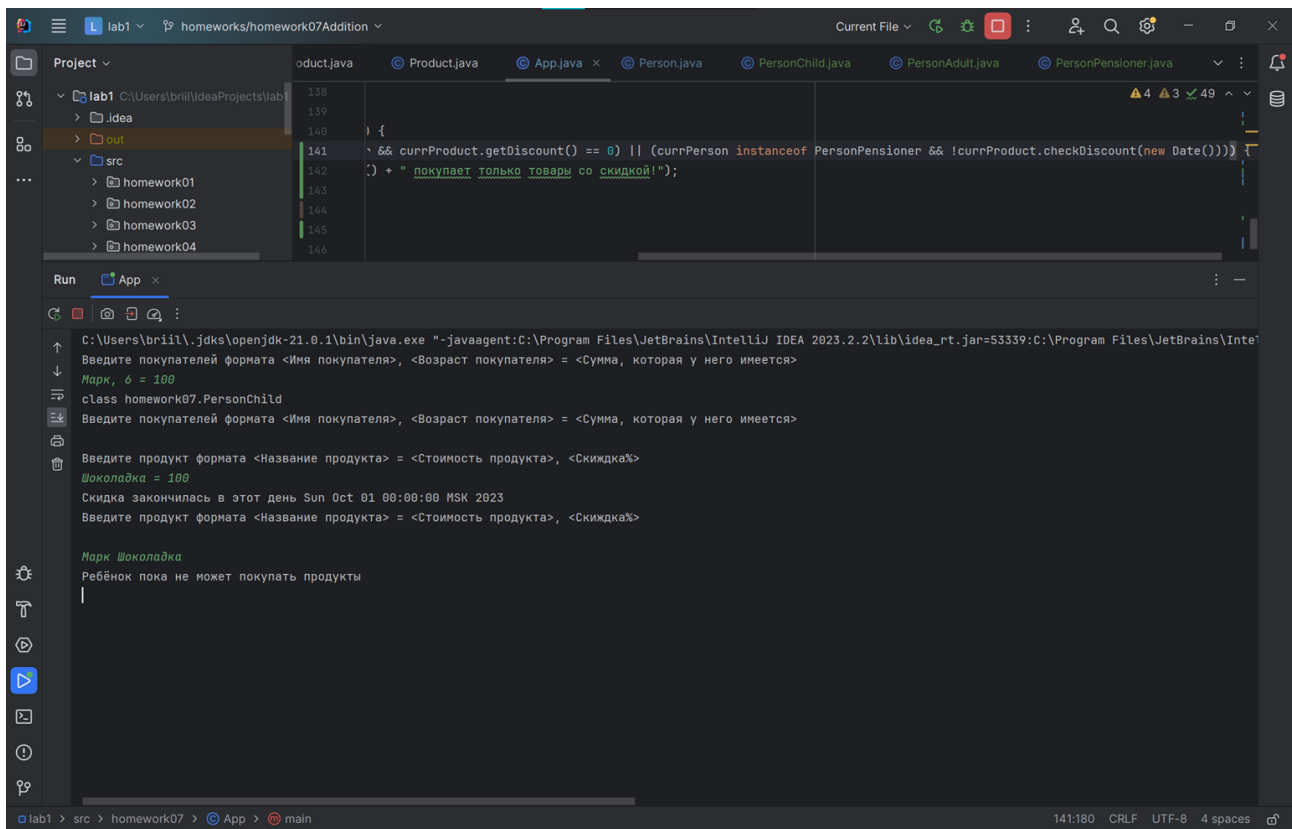
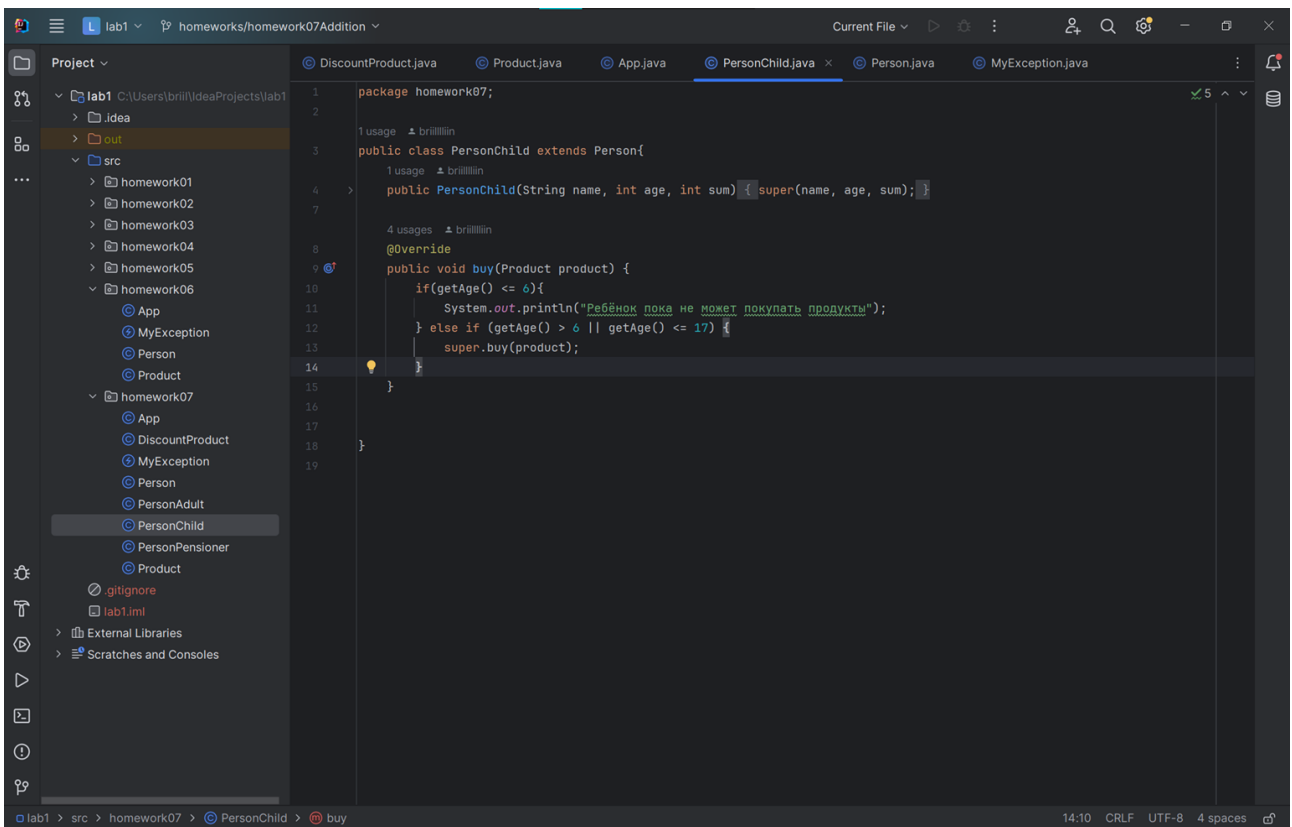


## Демонстрация работы в консоли

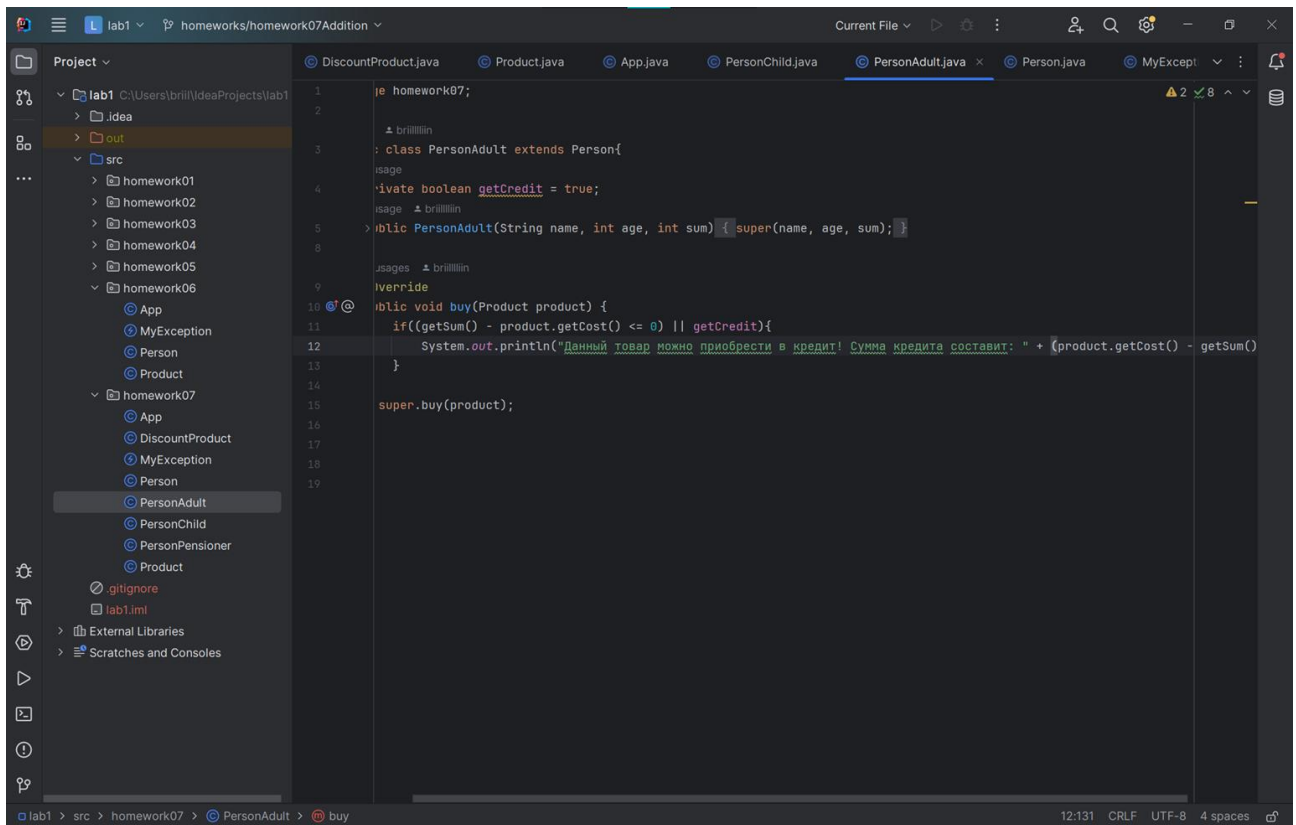




## Класс PersonChild

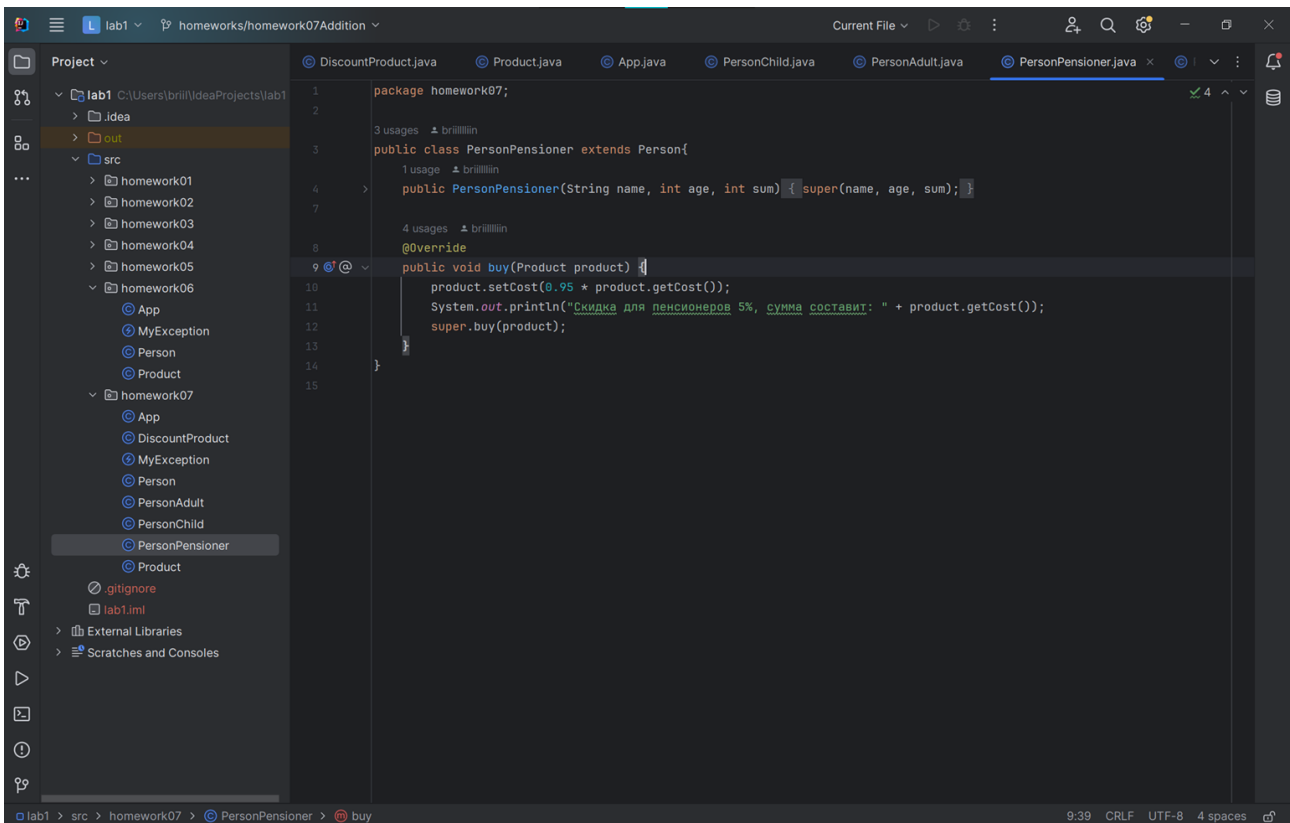


## Класс PersonAdult



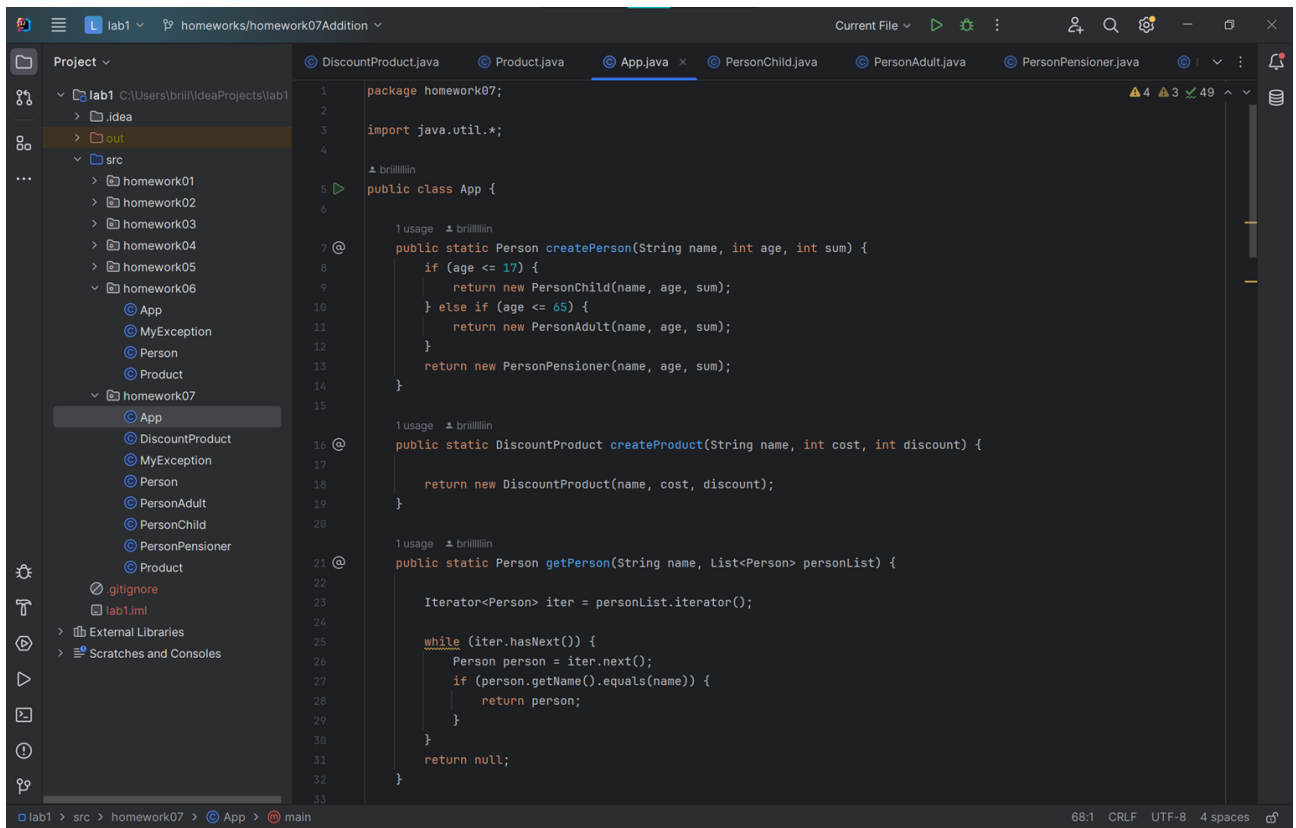
```
1  je homework07;
2
3  class PersonAdult extends Person{
4      private boolean getCredit = true;
5      public PersonAdult(String name, int age, int sum) { super(name, age, sum); }
6
7      @Override
8      public void buy(Product product) {
9          if((getSum() - product.getCost()) <= 0) || getCredit){
10             System.out.println("Данный товар можно приобрести в кредит! Сумма кредита составит: " + (product.getCost() - getSum()));
11         }
12         super.buy(product);
13     }
14 }
15
16
17
18
19
```

## Класс PersonPensioner

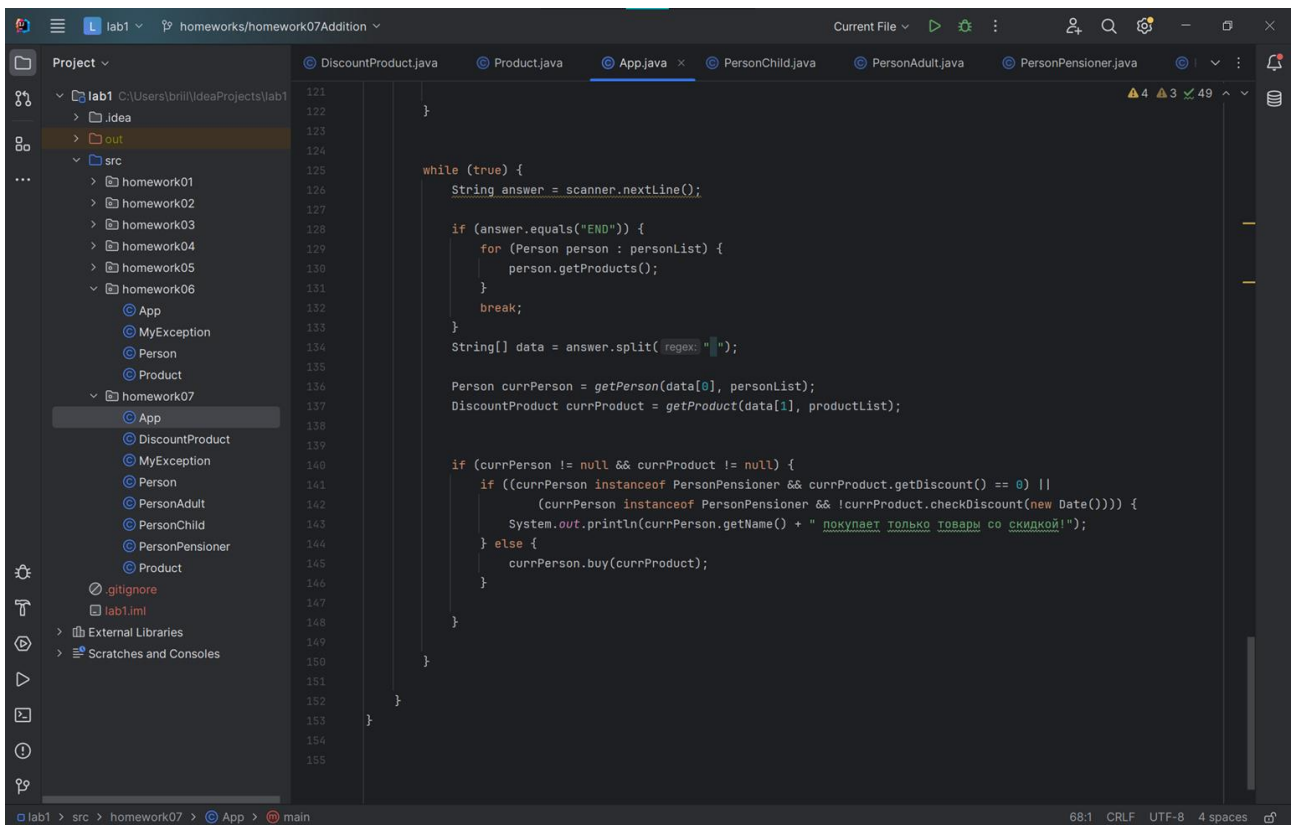


```
1  package homework07;
2
3  public class PersonPensioner extends Person{
4      public PersonPensioner(String name, int age, int sum) { super(name, age, sum); }
5
6      @Override
7      public void buy(Product product) {
8          product.setCost(0.95 * product.getCost());
9          System.out.println("Скидка для пенсионеров 5%, сумма составит: " + product.getCost());
10         super.buy(product);
11     }
12 }
13
14
15
```

## Основные изменения класса App, проверяющие работу с наследниками класса Person



```
1 package homework07;
2
3 import java.util.*;
4
5 public class App {
6
7     public static Person createPerson(String name, int age, int sum) {
8         if (age <= 17) {
9             return new PersonChild(name, age, sum);
10        } else if (age <= 65) {
11            return new PersonAdult(name, age, sum);
12        }
13        return new PersonPensioner(name, age, sum);
14    }
15
16    public static DiscountProduct createProduct(String name, int cost, int discount) {
17
18        return new DiscountProduct(name, cost, discount);
19    }
20
21    public static Person getPerson(String name, List<Person> personList) {
22
23        Iterator<Person> iter = personList.iterator();
24
25        while (iter.hasNext()) {
26            Person person = iter.next();
27            if (person.getName().equals(name)) {
28                return person;
29            }
30        }
31        return null;
32    }
33 }
```



```
121 }
122
123
124
125 while (true) {
126     String answer = scanner.nextLine();
127
128     if (answer.equals("END")) {
129         for (Person person : personList) {
130             person.getProducts();
131         }
132         break;
133     }
134     String[] data = answer.split(" ");
135
136     Person currPerson = getPerson(data[0], personList);
137     DiscountProduct currProduct = getProduct(data[1], productList);
138
139     if (currPerson != null && currProduct != null) {
140         if ((currPerson instanceof PersonPensioner && currProduct.getDiscount() == 0) ||
141             (currPerson instanceof PersonPensioner && !currProduct.checkDiscount(new Date()))) {
142             System.out.println(currPerson.getName() + " покупает только товары со скидкой!");
143         } else {
144             currPerson.buy(currProduct);
145         }
146     }
147 }
148
149
150 }
151
152
153 }
154
155 }
```