Модуль на Java

```java
package Ticket;

import java.time.LocalDate;

import static java.time.temporal.ChronoUnit.DAYS;

public class  returnTicketClass {
    public static double returnTicket(LocalDate date, int price){
        if(price <= 0) {
            System.out.println("Введена некорректная цена");
        }

        LocalDate currentDate = LocalDate.now();

        long difference =  DAYS.between(currentDate, date);

        if( (difference <= 5) && difference > 3){
            price = (price / 100) * 30;
        }else if((difference <= 10) && difference > 5){
            price = (price / 100) * 50;
        }else if(difference >= 10){
            price = price;
        }else if(difference < 3) {
            price = 0;
        }
        return price;
    }
}
```

Составить не менее 5 автоматизированных тестов с позитивным сценарием

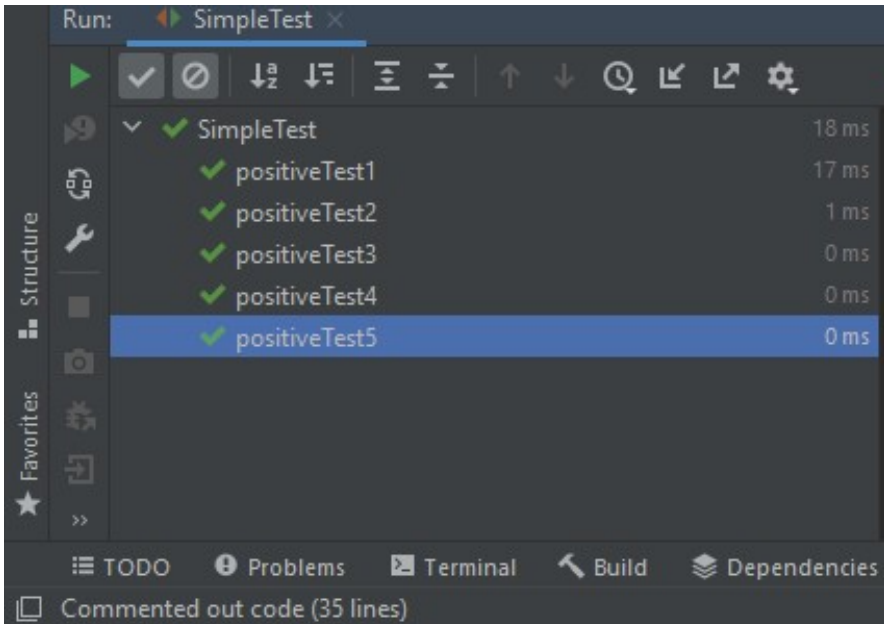
```java
public class SimpleTest {

    // Автотесты с позитивным сценарием
    @Test
    public void positiveTest1()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 06, dayOfMonth: 10),  price: 2500);
        Assert.assertEquals( expected: 2500, actualResult,  delta: 0);
    }

    @Test
    public void positiveTest2()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 05, dayOfMonth: 29),  price: 4200);
        Assert.assertEquals( expected: 0, actualResult,  delta: 0);
    }

    @Test
    public void positiveTest3()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 06, dayOfMonth: 2),  price: 10000);
        Assert.assertEquals( expected: 3000, actualResult,  delta: 0);
    }

    @Test
    public void positiveTest4()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 06, dayOfMonth: 5),  price: 300);
        Assert.assertEquals( expected: 150, actualResult,  delta: 0);
    }
    @Test
    public void positiveTest5()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2023, month: 12, dayOfMonth: 31),  price: 1000);
        Assert.assertEquals( expected: 1000, actualResult,  delta: 0);
    }
```



Составить не менее 5 автоматизированных тестов с негативным сценарием

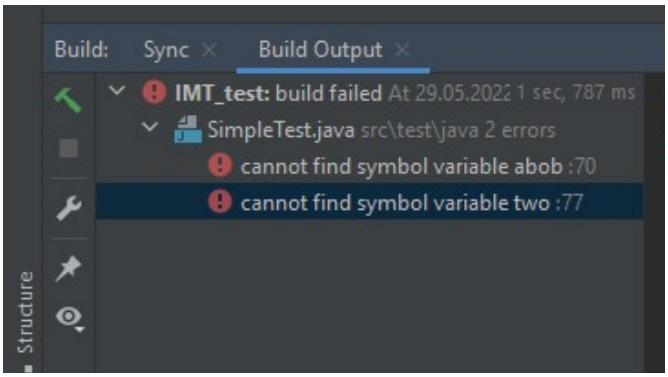
```java
    // Автотесты с негативным сценарием
    @Test
    public void negativeTest1()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 15, dayOfMonth: 22),  price: 900);
        Assert.assertEquals( expected: 900, actualResult,  delta: 0);
    }

    @Test
    public void negativeTest2()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 15, dayOfMonth: 22),  price: -900);
        Assert.assertEquals( expected: 900, actualResult,  delta: 0);
    }

    @Test
    public void negativeTest3()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 1922, month: 12, dayOfMonth: 22),  price: 1900);
        Assert.assertEquals( expected: 0, actualResult,  delta: 0);
    }

    @Test
    public void negativeTest4()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 12, dayOfMonth: 22), abob);
        Assert.assertEquals( expected: 0, actualResult,  delta: 0);
    }

    @Test
    public void negativeTest5()
    {
        double actualResult = returnTicketClass.returnTicket(LocalDate.of(2022,two,22),  price: 100);
        Assert.assertEquals( expected: 0, actualResult,  delta: 0);
    }
}
```



Составить 5 ручных тестов

```java
package com.company;

import java.time.LocalDate;

public class Main {

    public static void main(String[] args) {

        //test 1
        System.out.println("test1:");

        System.out.println(returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 06, dayOfMonth: 10),  price: 2500));

        // test2
        System.out.println("test2:");

        System.out.println(returnTicketClass.returnTicket(LocalDate.of( year: 2019, month: 06, dayOfMonth: 5),  price: 600));

        // test3
        System.out.println("test3:");

        System.out.println(returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 06, dayOfMonth: 5),  price: 1300));

        // test4
        System.out.println("test4:");

        System.out.println(returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 06, dayOfMonth: 10),  price: 0));

        // test5
        System.out.println("test5:");

        System.out.println(returnTicketClass.returnTicket(LocalDate.of( year: 2022, month: 13, dayOfMonth: 10),  price: 666));
    }
}
```

```
Main
D:\Java\bin\java.exe "-javaagent:D:\intellij\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=50150:D:\intellij\IntelliJ I
test1:
2500.0
test2:
0.0
test3:
650.0
test4:
Введена некорректная цена
0.0
test5:
Exception in thread "main" java.time.DateTimeException Create breakpoint : Invalid value for MonthOfYear (valid values 1 - 12): 13
    at java.base/java.time.temporal.ValueRange.checkValidValue(ValueRange.java:319)
    at java.base/java.time.temporal.ChronoField.checkValidValue(ChronoField.java:718)
    at java.base/java.time.LocalDate.of(LocalDate.java:271)
    at com.company.Main.main(Main.java:32)

Process finished with exit code 1
```