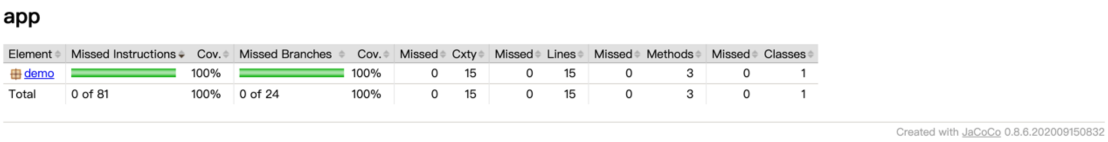
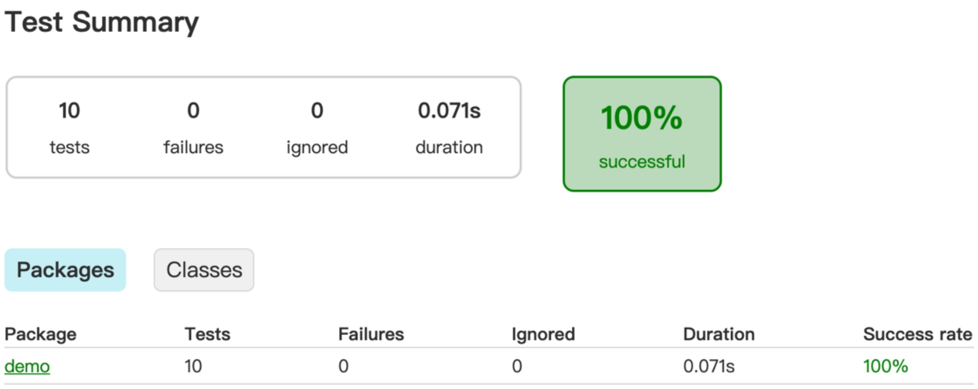


app

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
demo	🟩	100%	🟩	100%	0	15	0	15	0	3	0	1
Total	0 of 81	100%	0 of 24	100%	0	15	0	15	0	3	0	1

Created with JaCoCo 0.8.6.202009150832



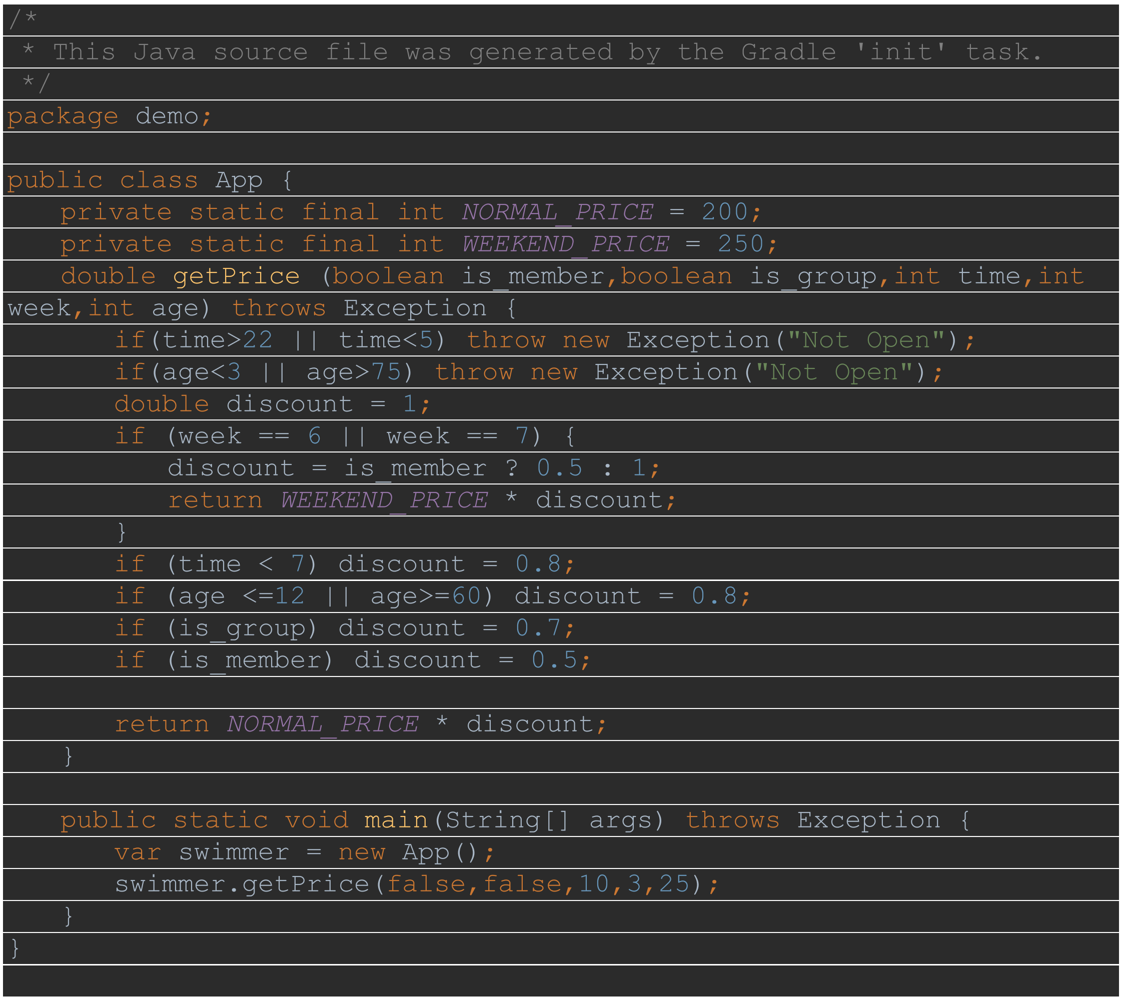
Test Summary

10 tests | 0 failures | 0 ignored | 0.071s duration

100% successful

Package	Tests	Failures	Ignored	Duration	Success rate
demo	10	0	0	0.071s	100%

App.java



```java
/*
 * This Java source file was generated by the Gradle 'init' task.
 */
package demo;

public class App {
    private static final int NORMAL_PRICE = 200;
    private static final int WEEKEND_PRICE = 250;
    double getPrice (boolean is_member,boolean is_group,int time,int
week,int age) throws Exception {
        if(time>22 || time<5) throw new Exception("Not Open");
        if(age<3 || age>75) throw new Exception("Not Open");
        double discount = 1;
        if (week == 6 || week == 7) {
            discount = is_member ? 0.5 : 1;
            return WEEKEND_PRICE * discount;
        }
        if (time < 7) discount = 0.8;
        if (age <=12 || age>=60) discount = 0.8;
        if (is_group) discount = 0.7;
        if (is_member) discount = 0.5;

        return NORMAL_PRICE * discount;
    }

    public static void main(String[] args) throws Exception {
        var swimmer = new App();
        swimmer.getPrice(false,false,10,3,25);
    }
}
```

AppTest.java

```java
/*
 * This Java source file was generated by the Gradle 'init' task.
 */
package demo;

import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class AppTest {
    App app;

    @BeforeEach
    void set(){
        app = new App();
    }
    @Test
    void runTest() throws Exception {
        App.main(null);
    }
    @Test
    void priceTestWeak1() throws Exception {
        assertEquals(100, app.getPrice(true,false,12,4,25));
        assertEquals(200, app.getPrice(false,false,12,4,25));
    }
    @Test
    void priceTestWeak2() throws Exception {
        assertEquals(140, app.getPrice(false,true,12,4,25));
        assertEquals(200, app.getPrice(false,false,12,4,25));
    }
    @Test
    void priceTestWeak3() throws Exception {
        assertEquals(160, app.getPrice(false,false,6,4,25));
        assertEquals(200, app.getPrice(false,false,12,4,25));
    }
    @Test
    void priceTestWeak4() throws Exception {
        assertEquals(250, app.getPrice(false,false,12,6,25));
        assertEquals(200, app.getPrice(false,false,12,4,25));
    }
    @Test
    void priceTestWeak5() throws Exception {
        assertEquals(160, app.getPrice(false,false,12,4,10));
        assertEquals(200, app.getPrice(false,false,12,4,25));
    }
    @Test
    void priceTestStrong1() {
        assertThrows(Exception.class, ()-> {
            app.getPrice(false, false, 4, 4, 25);
        });
        assertThrows(Exception.class, ()-> {
            app.getPrice(false, false, 23, 4, 25);
        });

    }
    @Test
    void priceTestStrong2() {
        assertThrows(Exception.class, ()-> {
            app.getPrice(false, false, 12, 4, 2);
        });
        assertThrows(Exception.class, ()-> {
            app.getPrice(false, false, 12, 4, 76);
        });

    }
    @Test
    void priceTestStrong3() throws Exception {
        assertEquals(125, app.getPrice(true,false,12,6,25));
        assertEquals(250, app.getPrice(false,true,12,6,25));
        assertEquals(250, app.getPrice(false,false,6,7,25));
        assertEquals(250, app.getPrice(false,false,12,7,10));
    }
    @Test
    void priceTestStrong4() throws Exception {
        assertEquals(160, app.getPrice(false,false,12,4,60));
    }
}
```