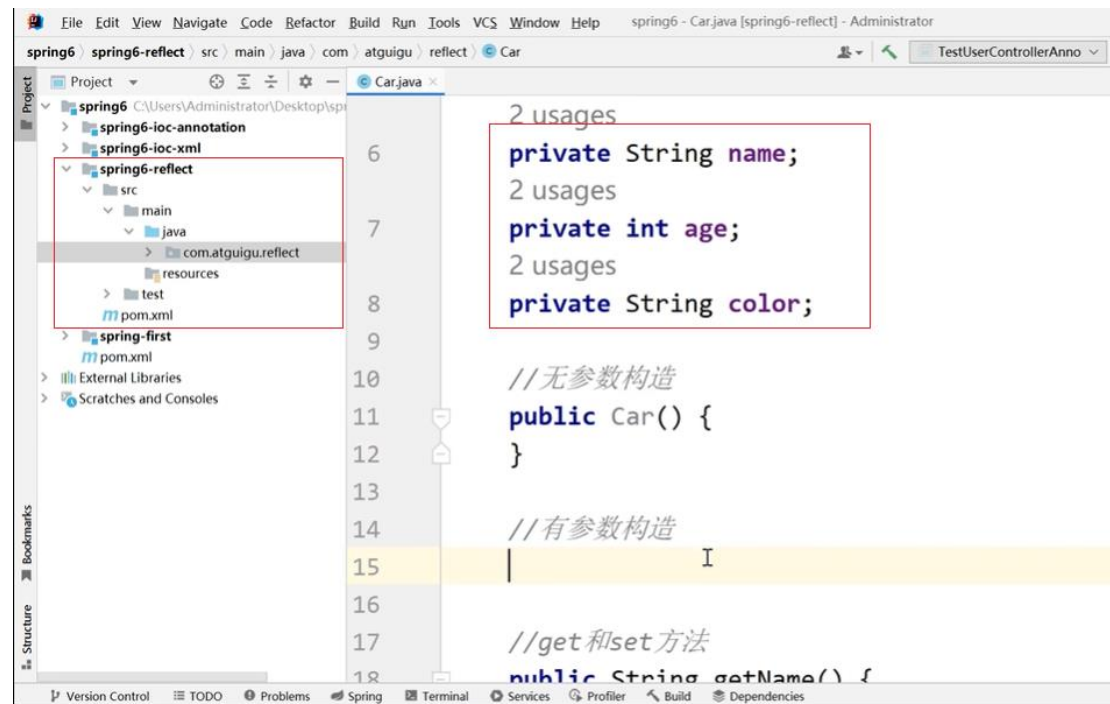


环境准备：

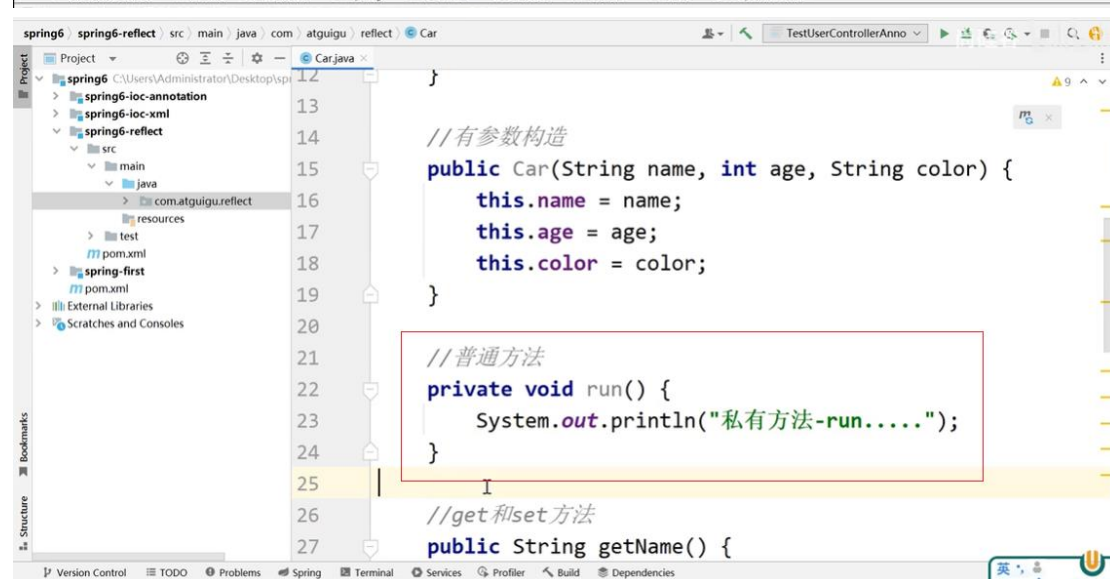


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - Car.java [spring6-reflect] - Administrator
spring6 spring6-reflect src main java com atguigu reflect Car
Project
  spring6 CAUsers\Administrator\Desktop\spring6
    spring6-annotation
    spring6-ioc-xml
    spring6-reflect
      src
        main
          java
            com.atguigu.reflect
              resources
                test
                  pom.xml
            spring-first
              pom.xml
  External Libraries
  Scratches and Consoles
Bookmarks
Structure
Car.java
6
7
8
9
10
11
12
13
14
15
16
17
18
2 usages
private String name;
2 usages
private int age;
2 usages
private String color;

//无参数构造
public Car() {
}

//有参数构造

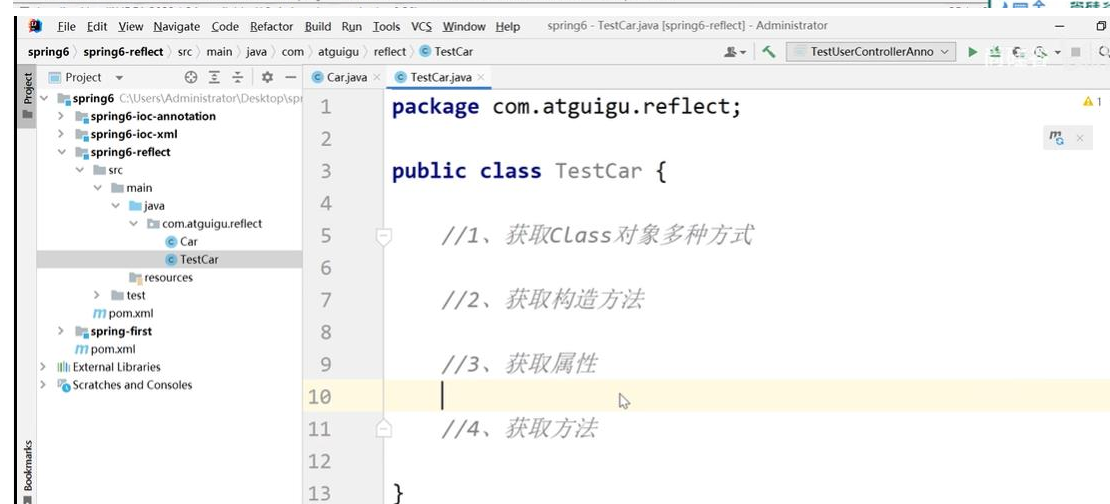
//get和set方法
public String getName() {
```



```
spring6 spring6-reflect src main java com atguigu reflect Car
Project
  spring6 CAUsers\Administrator\Desktop\spring6
    spring6-annotation
    spring6-ioc-xml
    spring6-reflect
      src
        main
          java
            com.atguigu.reflect
              resources
                test
                  pom.xml
            spring-first
              pom.xml
  External Libraries
  Scratches and Consoles
Bookmarks
Structure
Car.java
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
//有参数构造
public Car(String name, int age, String color) {
    this.name = name;
    this.age = age;
    this.color = color;
}

//普通方法
private void run() {
    System.out.println("私有方法-run.....");
}

//get和set方法
public String getName() {
```



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - TestCar.java [spring6-reflect] - Administrator
spring6 spring6-reflect src main java com atguigu reflect TestCar
Project
  spring6 CAUsers\Administrator\Desktop\spring6
    spring6-annotation
    spring6-ioc-xml
    spring6-reflect
      src
        main
          java
            com.atguigu.reflect
              Car
              TestCar
              resources
                test
                  pom.xml
            spring-first
              pom.xml
  External Libraries
  Scratches and Consoles
Bookmarks
Structure
TestCar.java
1
2
3
4
5
6
7
8
9
10
11
12
13
package com.atguigu.reflect;

public class TestCar {

    //1、获取Class对象多种方式
    //2、获取构造方法
    //3、获取属性
    //4、获取方法
}
```

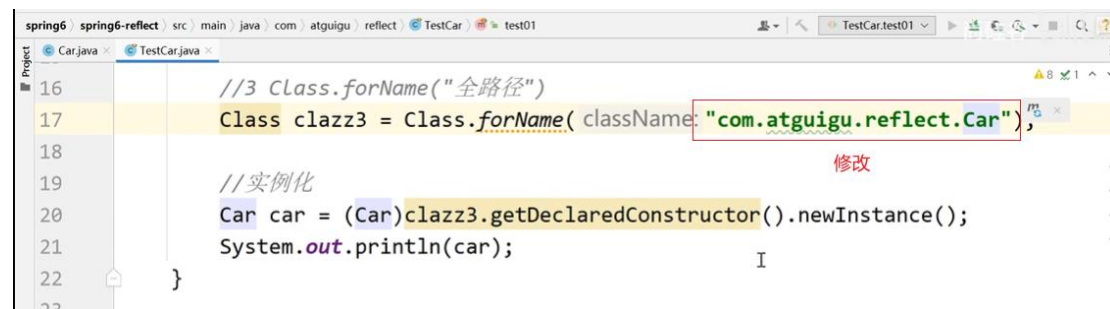
1、

```
//1、获取Class对象多种方式
@Test
public void test01() throws Exception {
    //1 类名.class
    Class clazz1 = Car.class;

    //2 对象.getClass()
    Class clazz2 = new Car().getClass();

    //3 Class.forName("全路径")
    Class clazz3 = Class.forName(className: "com.atguigu.reflect");
}
```

2、实例化



```
spring6 spring6-reflect src main java com atguigu reflect TestCar test01
16 //3 Class.forName("全路径")
17 Class clazz3 = Class.forName(className: "com.atguigu.reflect.Car");
18
19 //实例化
20 Car car = (Car)clazz3.getDeclaredConstructor().newInstance();
21 System.out.println(car);
22
23 }
```

3、



```
spring6 spring6-reflect src main java com atguigu reflect TestCar test02
26 //2、获取构造方法
27 @Test
28 public void test02() throws Exception {
29     Class clazz = Car.class;
30     // 获取所有构造
31     // getConstructors() 获取所有public的构造方法
32     //
33     Constructor[] constructors = clazz.getConstructors();
34     Constructor[] constructors = clazz.getDeclaredConstructors();
35     for (Constructor c:constructors) {
36         System.out.println("方法名称: "+c.getName()+" 参数个数: "+c.getParameters().length);
37     }
38
39 }
```

4、

```
spring6 spring6-reflect src main java com atguigu reflect TestCar test02
Car.java TestCar.java
38
39 //指定有参数构造创建对象
40 //1 构造public
41 // Constructor c1 = clazz.getConstructor(String.class, int.class, String.class);
42 // Car car1 = (Car)c1.newInstance("夏利", 10, "红色");
43 // System.out.println(car1);
44
45 //2 构造private
46 Constructor c2 = clazz.getDeclaredConstructor(String.class, int.class, String.class);
47 c2.setAccessible(true);
48 Car car2 = (Car)c2.newInstance("捷达", 15, "白色");
49 System.out.println(car2);
50 }
51
```

5、

```
spring6 spring6-reflect src main java com atguigu reflect TestCar test03
Car.java TestCar.java
52
53 //3、获取属性
54 @Test
55 public void test03() throws Exception {
56     Class clazz = Car.class;
57     Car car = (Car)clazz.getDeclaredConstructor().newInstance();
58     //获取所有public属性
59     //Field[] fields = clazz.getFields();
60     //获取所有属性（包含私有属性）
61     Field[] fields = clazz.getDeclaredFields();
62     for (Field field:fields) {
63         if(field.getName().equals("name")) {
64             //设置允许访问
65             field.setAccessible(true);
66             field.set(car, "五菱宏光");
67         }
68         System.out.println(field.getName());
69         System.out.println(car);
70     }
71 }
```

6、

```

75      @Test
76      public void test04() throws Exception {
77          Car car = new Car( name: "奔驰", age: 10, color: "黑色");
78          Class clazz = car.getClass();
79          //1 public方法
80          Method[] methods = clazz.getMethods();
81          for (Method m1:methods) {
82              //System.out.println(m1.getName());
83              //执行方法 toString
84              if(m1.getName().equals("toString")) {
85                  String invoke = (String)m1.invoke(car);
86                  System.out.println("toString执行了: "+invoke);
87              }
88          }
89      }

```

7、

```

spring6 spring6-reflect src main java com atguigu reflect TestCar test04
Project Car.java TestCar.java
87      }
88      }
89      }
90      //2 private方法
91      Method[] methodsAll = clazz.getDeclaredMethods();
92      for (Method m:methodsAll) {
93          //执行方法 run
94          if(m.getName().equals("run")) {
95              m.setAccessible(true);
96              m.invoke(car);
97          }
98      }
99      }
100     }

```