

Московский Авиационный Институт  
(Национальный Исследовательский Университет)

## Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

### Лабораторная работа 8 по курсу ООП: основы программирования на языке C#

#### 8. КОНКРЕТИЗАЦИЯ

Работу выполнил:

М8О-205Б-21 Титеев Р.М.

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(вариант)

Руководитель: \_\_\_\_\_/Кузнецова С.В.

(подпись)

Дата: \_\_\_\_ октября 2022

# Конкретизация

## Текст программы

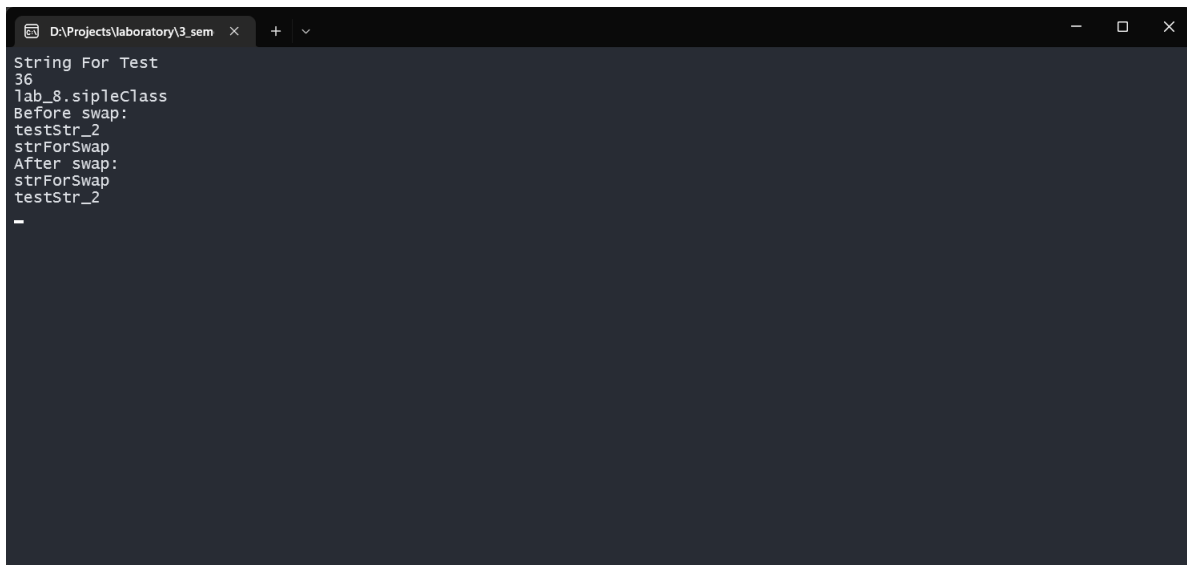
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace lab_8
8 {
9     class simpleClass {
10         public simpleClass(){}
11     }
12     class mainClass_1 <T> {
13         public mainClass_1(){}
14         public void m(T t){
15             Console.WriteLine(t.ToString());
16         }
17         public void m(){
18             Console.WriteLine(val.ToString());
19         }
20         public T val { set; get; }
21     }
22
23     class mainClass_2 <T1, T2, T3> {
24         public mainClass_2(){}
25         public void m(T2 t2) { Console.WriteLine(t2.ToString());}
26         public void m(T3 t3) { Console.WriteLine(t3.ToString());}
27         public void swap(ref T1 t1, ref T1 t2){
28             T1 temp = t1;
29             t1 = t2;
30             t2 = temp;
31         }
32         public T1 t1 { set; get;}
33         public T2 t2 { set; get;}
34         public T3 t3 { set; get;}
35     }
36
37     internal class Program
38     {
39         static void Main(string[] args)
40         {
41             mainClass_1<string> a_str = new mainClass_1<string>();
42             a_str.m("String For Test");
43
44             mainClass_1<int> a_int = new mainClass_1<int>();
```

```

45         a_int.m(4*9);
46
47         mainClass_1<simpleClass> a_simple = new mainClass_1<simpleClass>()
48         ;
49         a_simple.m(new simpleClass());
50
51         mainClass_2 <mainClass_1<string>,
52             mainClass_1<simpleClass>,
53             simpleClass> a_ = new mainClass_2<mainClass_1<string
54             >,
55                                     mainClass_1<
56                                     simpleClass>,
57                                     simpleClass>();
58
59         a_.t3 = new simpleClass();
60
61         a_str.val = "testStr_2";
62         var v = new mainClass_1<string>();
63         v.val = "strForSwap";
64
65         Console.WriteLine("Before swap:");
66         a_str.m();
67         v.m();
68
69         a_.swap(ref a_str, ref v);
70
71         Console.WriteLine("After swap:");
72         a_str.m();
73         v.m();
74
75         Console.ReadKey();
76     }
77 }
78

```

## Результат работы



```
String For Test
36
lab_8.sipleClass
Before swap:
testStr_2
strForSwap
After swap:
strForSwap
testStr_2
-

```

## Вывод

Конкретизация позволяет описывать классы и функции для работы с объектами, типы которых заранее не определены. С каким(и) именно типам(и) придется работать определяется в момент вызова – имя типа настраивает функцию или класс для работы.