# Упражнения: Класове и обекти

Можете да тествате решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/3933>.

## Клас „квадрат“

**Важно:** поради технически особености на SoftUni Judge трябва да имате **публичен клас StartUp** в namespace \_01\_ClassSquare. Кодът ви трябва да е структуриран по следния начин:

Graphical user interface

Description automatically generated with low confidence

Дефинирайте клас **Square**, който има **едно свойство** - **Side** (int). Добавете **getter** и **setter**. В класа **StartUp** създайте инстанция на класа **Square** с размер на страната **6**. Отпечатайте неговата **обиколка** и **лице**.

### Насоки

1. В namespace-a **SquareCalculator** добавете клас **Square**.
   1. В панела **Solution** натиснете **десен бутон** на мишката върху името на класа

Graphical user interface, text, application

Description automatically generated

* 1. От падащото меню селектирайте **Add -> Class**

Graphical user interface, application

Description automatically generated

* 1. В отворения прозорец селектирайте опцията **Empty Class**, напишете името на класа – **Square** – и изберете **[Create]**.

Graphical user interface, application

Description automatically generated

1. Добавете класа в **namespace** **\_01\_ClassSquare**

Chart

Description automatically generated with low confidence

1. Добавете **частно поле side** и **публично свойство Side** с getter и setter

Text

Description automatically generated with low confidence

1. В класа **StartUp** създайте инстанция на класа **Square**

Graphical user interface, text, application

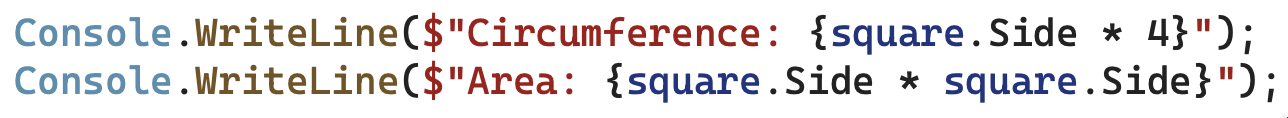
Description automatically generated

1. Задайте дължината на **страната на квадрата**, равна на **6**

A picture containing text, clipart, tableware, dishware

Description automatically generated

1. Отпечатайте **обиколката** (circumference) и **лицето** (area) на квадрата



**Изход** в терминала:

Graphical user interface, text, application, chat or text message

Description automatically generated

## Клас „триъгълник“

**Важно:** поради технически особености на SoftUni Judge трябва да имате **публичен клас StartUp** в namespace \_02\_Triangle. Кодът ви трябва да е структуриран по следния начин:

A picture containing application

Description automatically generated

Дефинирайте клас **Triangle**, който има **три свойства**: **SideA**, **SideB** и **SideC** (цели числа). Добавете   
**getter**-и и **setter-**и. В класа **StartUp** създайте инстанция на класа **Triangle** със страни **6**, **12** и **16**. неговата **обиколката** на триъгълника.

### Насоки

1. В namespace-a **TriangleCalculator** добавете клас **Triangle** със **свойства SideA**, **SideB** и **SideC**

Graphical user interface, text, application

Description automatically generated

1. В класа **StartUp** създайте инстанция на класа **Triangle** и задайте дължините на **страните на триъгълника**

Graphical user interface, text, application

Description automatically generated

1. Изчислете и отпечатайте **обиколката** на триъгълника във формата “**Circumference: {обиколка}**”

**Изход** в терминала:

Graphical user interface, text, application, chat or text message

Description automatically generated

## Клас „триъгълник“ с конструктор

Използвайте класа **Triangle** от миналата задача и добавете **конструктор**, който приема **трите страни** на триъгълника. За да го **тествате**, в **StartUp** създайте **нова инстанция** на класа **Triangle** и като параметри на конструктора подайте **3** (дължината на **sideA**), **4** (**sideB**) и **5** (**sideC**).

### Насоки

1. Добавете **конструктор** на класа **Triangle**

Graphical user interface, text, application

Description automatically generated

1. В класа **StartUp** създайте нова инстанция на **Triangle** и подайте като **параметри на конструктора** страните със стойности **3**, **4** и **5**.

Graphical user interface, text, application

Description automatically generated

1. Изчислете и отпечатайте **обиколката** на триъгълника

**Изход** в терминала:

Graphical user interface, text, application, chat or text message

Description automatically generated

## Клас „триъгълник“ с метод

Към класа **Triangle** от предишната задача добавете **метод calcCircumference()**, който **изчислява** и **връща обиколката** на триъгълника.

### Насоки

1. В класа **Triangle** добавете **метод calcCircumference()**, който да **изчисли** и да **върне** **обиколката** на триъгълника

A picture containing chart

Description automatically generated

1. В класа **StartUp** запазете създадената инстанция, след това извикайте върху нея метода **CalcCircumference** и отпечатайте неговия резултат

Graphical user interface, text, application

Description automatically generated

**Изход** в терминала:

Graphical user interface, text, application

Description automatically generated

## Клас „точка“

Създайте клас **Point**, който да има **два конструктора**:

**Първият** конструктор приема следните **параметри**:

* **X** –цяло число – **x-координатата** на точката
* **Y** –цяло число – **y-координатата** на точката

**Вторият** конструктор приема следните **параметри**:

* **X** –цяло число – **x-координатата** на точката
* **Y** –цяло число – **y-координатата** на точката
* **name** – стринг – **име** (най-често буквено означение) на точката

Класът **Point** трябва да има следните методи:

* **CalcDistance**(Point **otherPoint**) – приема **инстанция** на класа **Point** и връща **дистанцията (разстоянието)** между **двете точки**
* **Print()** – отпечатва информацията за точката в следния формат:
  + Ако точката **има име**: **“{name} ({x}, {y})”** – например **P1 (20, 30)**
  + Ако точката **няма име**: **“({x}, {y})”** – например **(20, 30)**

Трябва да може да използвате класовете по следния начин:

Graphical user interface, text, application

Description automatically generated

**Изход** в терминала:

Text

Description automatically generated