# Упражнения: Абстрактни класове и интерфейси

Можете да проверите решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/4066>

Състояние на настоящия учебен материал:



## Фигури

**Важно**: Трябва да имате публичен клас StartUp в namespace **\_01\_Shapes**.

Създайте **йерархия** от **интерфейси** и **класове**:



И двата класа трябва да отпечатват фигурата с "**\***".

Трябва да можете да използвате класа по следния начин:

static void Main(string[] args)

{

int radius = int.Parse(Console.ReadLine());

IDrawable circle = new Circle(radius);

int width = int.Parse(Console.ReadLine());

int height = int.Parse(Console.ReadLine());

IDrawable rectangle = new Rectangle(width, height);

circle.Draw();

rectangle.Draw();

}

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  4  5 | \*\*\*\*\*\*\*  \*\* \*\*  \*\* \*\*  \* \*  \*\* \*\*  \*\* \*\*  \*\*\*\*\*\*\*  \*\*\*\*  \* \*  \* \*  \* \*  \*\*\*\* |

### Решение

Алгоритъмът за рисуване на кръг е:

double radiusIn = this.radius - 0.4;

double radiusOut = this.radius + 0.4;

for (double y = this.radius; y >= -this.radius; y--)

{

for (double x = -this.radius; x < radiusOut; x += 0.5)

{

double value = x \* x + y \* y;

if (value >= radiusIn \* radiusIn && value <= radiusOut \* radiusOut)

Console.Write('\*');

else

Console.Write(' ');

}

Console.WriteLine();

}

Алгоритъмът за рисуване на правоъгълник е:

public void Draw()

{

DrawLine(this.width, '\*', '\*');

for (int i = 1; i < this.height - 1; i++)

DrawLine(this.width, '\*', ' ');

DrawLine(this.width, '\*', '\*');

}

private void DrawLine(int width, char end, char mid)

{

Console.Write(end);

for (int i = 1; i < width - 1; i++)

Console.Write(mid);

Console.WriteLine(end);

}

## Коли

**Важно**: Трябва да имате публичен клас StartUp в namespace Cars.

Създайте **йерархия** от **интерфейси** и **класове**:

Diagram

Description automatically generated

Трябва да можете да използвате класа по следния начин:

Graphical user interface, text

Description automatically generated

### Примери

|  |
| --- |
| **Изход** |
| Grey Renault Duster  Engine start  Breaaak!  Red Tesla Model 3 with 2 Batteries Engine start  Breaaak! |

## Дефинирайте интерфейс IPerson

**Важно**: Трябва да имате публичен клас StartUp в namespace PersonInfo.

Дефинирайте **интерфейс IPerson** със свойства за **Name** и **Age**. Дефинирайте клас **Citizen**, който имплементира **IPerson** и има **конструктор**, който приема:

* **name – string**
* **age - int**

Създайте нов **Person** по следния начин:

Text

Description automatically generated

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Pesho  25 | Pesho  25 |

## Множество имплементации

**Важно**: Трябва да имате публичен клас StartUp в namespace PersonInfo.

Използвайки кода от предишната задача, дефинирайте **интерфейс IIdentifiable** със свойство **string Id** и **интерфейс IBirthable** със свойство **string Birthdate**. Имплементирайте ги в класа **Citizen**. Презапишете конструктора на **Citizen**, за да приема новите параметри.

Трябва да можете да използвате класа по следния начин:

Graphical user interface, text, application

Description automatically generated

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Pesho  25  9105152287  15/05/1991 | 9105152287  15/05/1991 |

## Телефония

Имате бизнес – **производство** на **телефони**. Само че нямате софтуерни инженери, така че се обаждате на приятели и ги молите за помощ при създаването на софтуер. Те се съгласяват и започвате да работите по проекта. Той се състои от **два главни модела** - **Smartphone** и **StationaryPhone**. Всеки смартфон трябва да има функционалности да **звъни на други телефони** и да **достъпва интернет**. **StationaryPhone** могат **само да звънят на други телефони**.

Оказва се, че приятелите ви са много заети, така че решавате да напишете кода сами. Изискванията са следните:

Трябва да имате **модел** **Smartphone** и две отделни функционалности на смартфона ви – да **звъни на други телефони** и да **достъпва интернет**.

Трябва да имате и **модел StationaryPhone** и функционалност, която има – да **звъни на други телефони**.

Трябва да имате **два класа** и **два интерфейса**.

### Вход

* Първи ред: **телефонни номера**, на които да се обаждате (**string**), разделени с интервал.
* Втори ред: **сайтове**, които да посетите (**string**), разделени с интервал.

### Изход

* Първо **звъннете на всички телефонни номера** в последователността, в която сте ги получили,а след това отворете **всички сайтове** в оригиналната последователност.
* Функционалността за звънене трябва да отпечатва на конзолата на кой номер звъните:
* Ако номерът е дълъг 10 цифри, звъните от смартфона и отпечатвате: **Calling... {number}**
* Ако номерът е дълъг 7 цифри, звъните от стационарния телефон и отпечатвате: **Dialing... {number}**
* Функционалността с браузъра трябва да отпечатва кой сайт е отворен:

**Browsing:** **{site}!**

* Ако има число в някой от URL-ите на сайтовете, отпечатайте **"Invalid URL!"** и продължете с останалите сайтове.
* Ако в някой телефонен номер има символ, различен от цифра, отпечатайте **"Invalid number!"** и продължете с останалите номера

### Бележки

* Телефонните номера винаги ще бъдат с дължина от 7 или 10 символа.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 0882134215 0882134333 0899213421 0558123 3333123  http://softuni.bg http://youtube.com http://www.g00gle.com | Calling... 0882134215  Calling... 0882134333  Calling... 0899213421  Dialing... 0558123  Dialing... 3333123  Browsing: http://softuni.bg!  Browsing: http://youtube.com!  Invalid URL! |