# Упражнения: Капсулация

Можете да тествате решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/4062>

Състояние на настоящия учебен материал:

A green and blue rectangle with white text

Description automatically generated

## Отпечатване на информация за човек

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

Graphical user interface

Description automatically generated with low confidence

Създайте **публичен** клас **Person**, който трябва да има **публични свойства** и **частни полета** за:

* **FirstName**: стринг (**string**)
* **LastName**: стринг (**string**)
* **Age**: цяло число (**int**)

Добавете **конструктор**, който приема **три параметъра** – **първо** и **фамилно** име и **възраст**.

Добавете следния **метод**:

* **ToString()**: стринг (**string**) **– override**
  + Връща информация за човека в следния формат:
    - **"{първо име} {фамилно име} is {възраст} years old."**

Трябва да можете да ползвате класа по следния начин:

static void Main(string[] args)

{

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

List<Person> people = new List<Person>();

for (int curLine = 1; curLine <= lines; curLine++)

{

List<string> tokens = Console.ReadLine().Split().ToList();

string firstName = tokens[0];

string lastName = tokens[1];

int age = int.Parse(tokens[2]);

Person person = new Person(firstName, lastName, age);

people.Add(person);

}

people.OrderBy(p => p.FirstName)

.ThenBy(p => p.Age)

.ToList()

.ForEach(p => Console.WriteLine(p));

}

Методът **.OrderBy()** се използва за **сортиране** на обектите по даден критерий (в случая по първото име). Ако искаме да добавим допълнителен критерий, използваме **.ThenBy()**.

За да работим с колекцията в удобен за нас формат, обръщаме резултата от сортирането в **списък** с **.ToList()**.

Методът **.ForEach()** в C# е еквивалентен на цикъла **foreach**. Чрез него можем да приложим дадена операция върху **всеки елемент** от списъкa – в случая взимаме всеки **p** (person) и го **отпечатваме** на конзолата.

Както вече знаете, при отпечатване с **Console.WriteLine()** метода автоматично върху обекта се **извиква** неговия **ToString()** метод.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 6  Seth Nelson 65  Liam Scott 57  Brian Clark 27  Alisa Bell 44  Brian White 12  Sophie Baker 35 | Alisa Bell is 44 years old.  Brian White 12 years old.  Brian Clark is 27 years old.  Liam Scott is 57 years old.  Seth Nelson is 65 years old.  Sophie Baker is 35 years old. |

### Насоки

1. Създайте **нов клас** и му задайте **коректно име**. Дефинирайте **публичните** свойства:

Table

Description automatically generated with medium confidence

1. Създайте конструктор за **Person**, който приема 3 параметъра - **firstName**, **lastName** и **age**:

Graphical user interface, text, application

Description automatically generated

1. Презапишете метода **ToString()**:

A picture containing logo

Description automatically generated

## Повишение на заплатата

**Важно**: Трябва да имате клас StartUp в namespace **PersonsInfo**.

Надграждайки кода от предишната задача, добавете следните **свойства** и **методи** към класа **Person**:

* **Salary**: **double**
* **IncreaseSalary**(**double** **percentage**) : **void**

Обновете **констурктора**, така че да приема като **параметър** и **заплатата**.

Методът **IncreaseSalary()** трябва да **обнови заплатата** на служителя спрямо въведения **процент**. Тези, които са **по-млади от 30 години**, получават **половината от повишението**.

Обновете метода **ToString()**, така че да връща информацията в следния формат:

* **"{първо име} {фамилно име} receives {повишена заплата} leva."**

Заплатата трябва да бъде форматирана **до втория знак** след десетичната точка.

Трябва да можете да използвате класа по следния начин:

static void Main(string[] args)

{

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

List<Person> people = new List<Person>();

for (int curLine = 1; curLine <= lines; curLine++)

{

List<string> tokens = Console.ReadLine().Split().ToList();

string firstName = tokens[0];

string lastName = tokens[1];

int age = int.Parse(tokens[2]);

double salary = double.Parse(tokens[3]);

Person person = new Person(firstName, lastName, age, salary);

people.Add(person);

}

double percentage = double.Parse(Console.ReadLine());

people.ForEach(p => p.IncreaseSalary(percentage));

people.ForEach(p => Console.WriteLine(p));

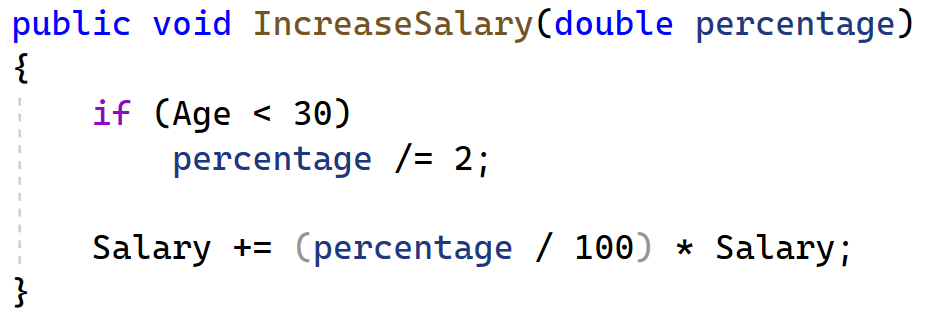
}

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5  Nick Adams 65 2200  Lynda Fisher 57 3333  Paul Walker 27 600  Vera Nelson 44 666.66  Connor Perry 35 559.4  20 | Nick Adams receives 2640.00 leva.  Lynda Fisher receives 3999.60 leva.  Paul Walker receives 660.00 leva.  Vera Nelson receives 799.99 leva.  Connor Perry receives 671.28 leva. |

### Насоки

1. Добавете ново **публично** свойство за **заплата** и рефакторирайте **конструктора**. Добавете нов **метод**, който да **обновява заплатата** с даден бонус:



2. Рефакторирайте метода **ToString()**.

## Валидация на данни

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

Разширете класа **Person** с **валидация** за всяко **поле**:

* **First** и **Last name** трябва да са с по **най-малко** **3 символа**
* **Agе** **не трябва** да бъде **нула** или **отрицателно число**
* **Заплатата** **не трябва** да бъде **по-малка от 460 (decimal)**

При **невалидни данни**, отпечатайте съответното съобщение:

* **"First name cannot contain fewer than 3 symbols!"**
* **"Last name cannot contain fewer than 3 symbols!"**
* **"Age cannot be zero or a negative integer!"**
* **"Salary cannot be less than 460 leva!"**

Използвайте **ArgumentExeption** за съобщенията.

Трябва да можете да използвате класа по следния начин:

static void Main(string[] args)

{

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

List<Person> people = new List<Person>();

for (int curLine = 1; curLine <= lines; curLine++)

{

List<string> tokens = Console.ReadLine().Split().ToList();

string firstName = tokens[0];

string lastName = tokens[1];

int age = int.Parse(tokens[2]);

double salary = double.Parse(tokens[3]);

try

{

Person person = new Person(firstName, lastName, age, salary);

people.Add(person);

}

catch (ArgumentException ex)

{

Console.WriteLine(ex.Message);

}

}

double percentage = double.Parse(Console.ReadLine());

people.ForEach(p => p.IncreaseSalary(percentage));

people.ForEach(p => Console.WriteLine(p));

}

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5  Miles Parks -6 2200  B Potter 57 3333  Julie Brown 27 600  Alice H 44 666.66  Joey Hall 35 300  20 | Age cannot be zero or a negative integer!  First name cannot contain fewer than 3 symbols!  Last name cannot contain fewer than 3 symbols!  Salary cannot be less than 460 leva!  Julie Brown gets 660.00 leva. |

## Клас Box

**Важно**: Трябва да имате публичен клас StartUp в namespace **\_05\_ClassBox**.

Дадена ви е кутия (правоъгълен паралелепипед) с параметри **length**, **width** и **height**. Създайте клас **Box**, който може да бъде инстанциран със същите **три параметъра**. Направете достъпни за външния свят **само свойствата**, **конструктора** и методитеза **лице на околната и пълната повърхнина** и за **обем.** (формули: <https://www.matematika.bg/geometry/volume.html>).

**Страната** на кутията **не трябва** да бъде **нула** или **отрицателно число.** Добавете **валидация на данните** за всекипараметър, подаден на конструктора. Направете **частен setter**, който осъществява **вътрешна валидация на данните.**

**П**ри въвеждане на невалидни данни, хвърлете **ArgumentException** със следното съобщение:

* **"{width/length/height} cannot be zero or negative."**

На **първите три реда** в **Main()** методаще получите **length** (дължина), **width** (ширина) и **height** (височина). На **следващите три реда** отпечатайте **пълната, околната повърхнина** и **обема** на кутията.

Ако се **хвърли** **Exception**, го **хванете** и отпечатайте неговото **съобщение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 2  3  4 | Surface Area - 52.00  Lateral Surface Area - 40.00  Volume - 24.00 |
| 1.3  1  6 | Surface Area - 30.20  Lateral Surface Area - 27.60  Volume - 7.80 |
| 2  -3  4 | Width cannot be zero or negative. |