# 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

## Клас Person: сортиране по име и възраст

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

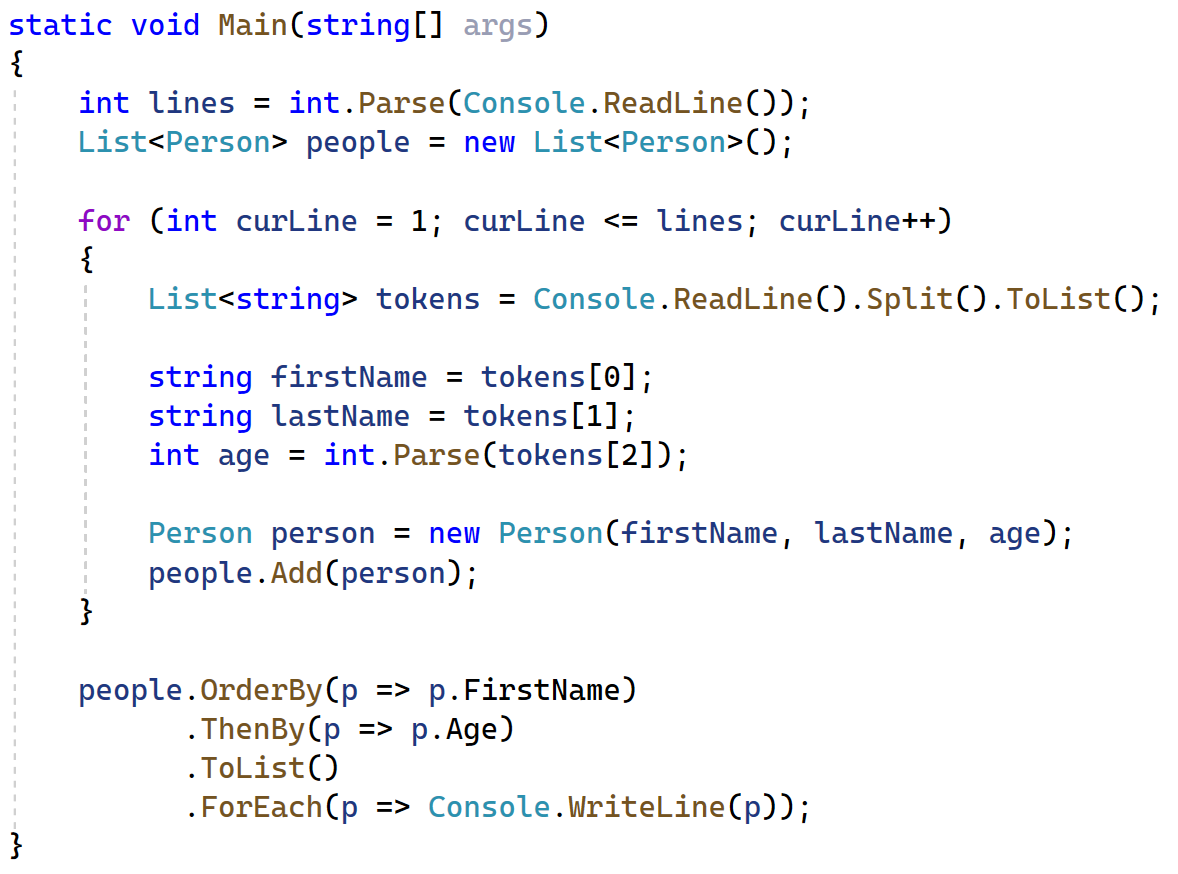
Graphical user interface

Description automatically generated with low confidence

Създайте клас **Person**, който трябва да има **публични** свойства и **частни** setter-и за:

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



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** |
| 5  Seth Nelson 65  Liam Scott 57  Brian Clark 27  Alisa Bell 44  Sophie Baker 35 | Alisa Bell is 44 years old.  Seth Nelson is 65 years old.  Sophie Baker is 35 years old.  Liam Scott is 57 years old.  Brian Clark is 27 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**.

Създайте обекти от класа **Person**. Прочетете техните **имена**, **възраст** и **заплата** от конзолата. Прочетете процента на **бонус заплатата** на всеки човек. Тези, които са **по-млади от 30 години**, получават **половината от повишението**. Разширете класа **Person** oт предишната задача.

Нови **свойства** и **методи:**

* **Salary**: **decimal**
* **IncreaseSalary**(**decimal** **percentage**)

Трябва да можете да използвате класа по следния начин:

Graphical user interface, text, application

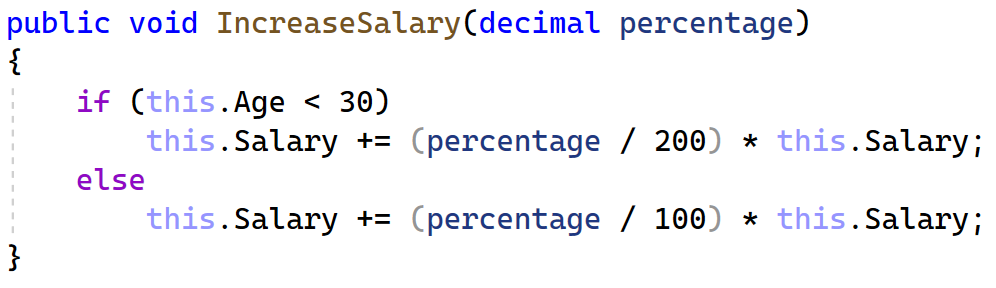
Description automatically generated

### 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.

### 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. |

### Насоки

1. Създайте **нов клас** и му задайте **коректно име**. Дефинирайте **публичните** свойства за **имена**, **възраст** и **заплата**.
2. Добавете **валидация** за полетата. При **невалидни данни** хвърлете **ArgumentationException** със съответното съобщение.
3. Презапишете метода **ToString()**.

## Главен и резервен екип

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

Създайте клас **Team**.

Класът трябва да има:

* **Частни полета** за:
  + **name**: **string**
  + **firstTeam**: List<Person>
  + **reserveTeam**: List<Person>
* **Публични свойства** за:
  + FirstTeam: List<Person> (read only!)
  + ReserveTeam: List<Person> (read only!)
* **Конструктор**
  + Team(string name)
* **Метод** за добавянена **играчи:**
  + AddPlayer(Person person): void

Добавете към отбора всички хора, които получавате. Тези, които са **по-млади от 40 години**,отиват в **първи екип** (first team), а останалите – в **резервния екип** (reserve team). Накрая отпечатайте **размера** на **първия** и на **резервния екип.**

Трябва да можете да използвате класа по следния начин:

Graphical user interface, text, application

Description automatically generated

**Не** трябвада можете да използвате класа по следния начин:

|  |
| --- |
| StartUp.cs |
| Team team = new Team("SoftUni");  foreach (Person person in persons)  {  if(person.Age < 40)  {  team.FirstTeam.Add(person);  }  else  {  team.ReserveTeam(person);  }  } |

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5  Troy Jones 20 2200  Martin Francis 57 3333  Ted Adams 27 600  Alisa Gomez 25 666.66  Lucia Cox 35 555 | First team has 4 players.  Reserve team has 1 players. |

## Клас Box

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

Дадена ви е кутия с параметри **length**, **width** и **height**. Създайте клас **Box**, който може да бъде инстанциран със същите **три параметъра**. Направете достъпни за външния свят **само 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.

### 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. |