

Limbajul: C#

Student propunător: Oarga Adriana Maria Grupa: 225 Email: adriimaria00@gmail.com

Ce se afișează la rularea codului urmator?

```
public interface Person
{
    String GiveGender();
    int Age { get; set; }
}

public class Woman : Person
{
    public int Age { get; set; }
    public string GiveGender(){ return "Feminin";}
}

public class Man : Person
{
    public int Age { get; set; }
    public string GiveGender(){return "Masculin";}
}

class Program
{
    public static object Statistics(List<Person> people)
    {
        return (from p in people where p.Age >= 18 select p);
    }
    static void Main(string[] args)
    {
        IList<Woman> women = new List<Woman>(){new Woman{Age = 16},new Woman{Age = 21}};
        IList<Man> men = new List<Man>() { new Man { Age = 68 },new Man {Age = 42} };
        var value = Statistics(women) < Statistics(men);
        Console.WriteLine($"There are more adult {(value is true? "men" : "women")}.");
    }
}
```

- Variante de raspuns:
  - a) There are more adult men.
  - b) syntax error : Two objects cannot be compared by operator '<'
  - c) runtime exception : Two objects cannot be compared by operator '<'
  - d) niciuna

- Justificare raspuns :

Metoda Statistics(IList<Person> people) cu aceasta signatura nu accepta parametrii de tipul claselor care implementeaza interfata Person ci doar lista obiecte de tipul Person. Solutia ar fi folosirea unei clase auxiliare sau sau a unei metode generice cu constrangeri care primeste tip generic T unde T implementeaza Person. Eroarea va fi de fapt syntax error (cannot do the cast between Person and Woman/Man)

- Concepte: LINQ, Wildcards, Generics, Inheritance
- Dificultate: Medium