UPskill – JAVA + .NET

Programação Orientada a Objetos - Classes e Objetos

Declaração e Uso de List

UPskill
Digital Skills & Jobs

- Uma interface é um tipo especial de declaração que lista um conjunto de métodos e suas assinaturas
  - Uma classe que implementa uma interface deve implementar todos os métodos da interface (Contrato)
  - Todos os métodos numa interface são abstratos (Sem implementação)
  - Para ajudar a identificar uma Interface usa-se o "I" no inicio do nome da mesma

```
public class Temperature : IComparable
{
    // The temperature value
    protected double temperatureF;

    0 references
```

Exemplo de Declaração de Interface

UPskill
Digital Skills & Jobs

```
public interface IComparable
{
    int CompareTo(object? obj);
}
```

```
5 references
public class Temperature : IComparable
{
    // The temperature value
    protected double temperatureF;

    0 references
    public int CompareTo(object obj)
    {
        if (obj == null)
        {
            return 1;
        }

        Temperature otherTemperature = obj as Temperature;
        if (otherTemperature != null)
        {
            return this.temperatureF.CompareTo(otherTemperature.temperatureF);
        }
        else
        {
            throw new ArgumentException("Object is not a Temperature");
        }
    }

    2 references
```

Interface IComparable

- Dot NET tal como o java inclui um conjunto de interfaces, entre elas a interface IComparable
  - É usada para comparação de dois objetos
  - Requer a implementação de um método: CompareTo()
  - Necessaria para uso de algumas funções tais como
    - Order

Interface IComparer (java Comparator)

UPskill
Digital Skills & Jobs

- Este método compara os objetos o1 e o2 e devolve um inteiro:
  - Negativo: o1 antecede o2
  - Zero: o1 é igual a o2
  - Positivo: o1 sucede o2

Exemplo IComparar

- Implementação da ordenação de contas bancárias, primeiro por owner (String), depois por balance (double)

```
4 references
public class BankAccount
{
    private double balance;
    private String owner;

    0 references
    public BankAccount(double bal, String name)
    {
        this.balance = bal;
        this.owner = name;
    }

    0 references
    public double getBalance()
    {
        return this.balance;
    }

    2 references
    public String getOwner()
    {
        return this.owner;
    }

    0 references
    public override string ToString()
    {
        return this.GetType().Name + ":" + this.balance
        + ", " + this.owner;
    }
```

Classe de implementa IComparer

- Implementação da classe BankSortbyOwnerComparator que implementa a interface Comparator, definindo o método compare()

```
0 references
public class BankSortbyOwnerComparator : IComparer<BankAccount>
{
    0 references
    public int Compare(BankAccount account1, BankAccount account2)
    {
        return account1.getOwner().CompareTo(account2.getOwner());
    }
}
```

Exemplo

UPskill
Digital Skills & Jobs

```
private static void Main(string[] args)
{
    var accounts = new List<BankAccount>();
    accounts.Add(new BankAccount(1000, "Jose"));
    accounts.Add(new BankAccount(2000, "Antonio"));
    accounts.Add(new BankAccount(500, "Manuel"));

    Console.WriteLine("Before sort:");
    foreach (var account in accounts)
    {
        Console.WriteLine(account);
    }

    accounts.Sort(new BankSortbyOwnerComparator());
    Console.WriteLine("After sort by owner:");
    foreach (var account in accounts)
    {
        Console.WriteLine(account);
    }
```