

Nas perguntas que seguem podem ser apresentados exemplos com utilização da classe Student.

```
class Student {  
    int nr;           // int é IComparable  
    string name;      // string é IComparable  
    [Comparison(typeof(AddressByStreet))] Address a; // Address NÃO é IComparable  
    [Comparison(typeof(AccountByBalance))] Account acc; // Account NÃO é IComparable  
    ...  
}
```

Dada a definição da interface IComparer, implemente o método **estático** Comparing, que retorna uma nova instância de IComparer cujo método Compare compara dois objectos e retorna um valor indicando quando um é menor, igual ou maior que o outro. (Recorra a API de System.Reflection).

```
interface IComparer { int Compare(object x, object y); }  
static IComparer Comparing(Type klass) { ... }
```

Use como critério de comparação os valores dos **campos** de klass que verifiquem uma das condições:

- Ser do tipo IComparable (interface IComparable{int CompareTo(object obj)}).
- Estar anotados com o *custom attribute* ComparisonAttribute

Implemente também o *custom attribute* ComparisonAttribute. Este atributo recebe o critério de comparação. Por exemplo para uma propriedade do tipo Address o critério de comparação podia ser definido com o comportamento:

```
((Address) x).street.CompareTo(((Address) y).street)
```

```
Student s1 = new Student(14000, "Ana", new Address("Rua Amarela", 24), new Account(9900));  
Student s2 = new Student(14000, "Ana", new Address("Rua Rosa", 24), new Account(9900));  
Student s3 = new Student(14000, "Ana", new Address("Rua Rosa", 24), new Account(100));  
IComparer cmp = Comparing(typeof(Student));  
int res1 = cmp.Compare(s1, s2); // res < 0 porque Rua Amarela é menor que Rua Rosa  
int res2 = cmp.Compare(s2, s3); // res > 0 porque 9900 é maior que 100
```