

```
using System;
```

```
class Evento
```

```
{
```

```
    // atributos do evento
```

```
    protected string Nome;
```

```
    protected string Local;
```

```
    protected string Horario;
```

```
    protected DateTime Data;
```

```
    protected double Preco;
```

```
    protected int ID;
```

```
    // métodos para acessar os atributos
```

```
    public string GetNome() { return Nome; }
```

```
    public string GetLocal() { return Local; }
```

```
    public string GetHorario() { return Horario; }
```

```
    public DateTime GetData() { return Data.Date; }
```

```
    public int GetID() { return ID; }
```

```
    public double GetPreco() { return Preco; }
```

```
    // métodos para modificar os atributos
```

```
    public void SetNome(string NovoNome) { Nome = NovoNome; }
```

```
    public void SetLocal(string NovoLocal) { Local = NovoLocal; }
```

```
    public void SetHorario(string NovoHorario) { Horario = NovoHorario; }
```

```
    public void SetData(DateTime NovaData) { Data = NovaData; }
```

```
    public void SetID(int NovoID) { ID = NovoID; }
```

```
    public void SetPreco(double NovoPreco) { Preco = NovoPreco; }
```

```
// construtor do evento
```

```
public Evento(int _ID, string _Nome, string _Local, DateTime _Data, string _Horario, double  
_Preco)
```

```
{
```

```
    ID = _ID;
```

```
    Nome = _Nome;
```

```
    Local = _Local;
```

```
    Horario = _Horario;
```

```
    Data = _Data;
```

```
    Preco = _Preco;
```

```
}
```

```
// compara eventos com base na escolha de ordenação
```

```
public bool Maior(Evento other, string resposta)
```

```
{
```

```
    bool result = false;
```

```
    // Ordena por ordem alfabetica
```

```
    if (resposta == "1" && other != null && Nome.CompareTo(other.Nome) > 0)
```

```
    {
```

```
        result = true;
```

```
    }
```

```
    // Ordena por ID
```

```
    if (resposta == "2" && other != null && ID > other.ID)
```

```
    {
```

```
        result = true;
```

```
    }
```

```
// Ordena por preço (menor para maior)
    if (resposta == "3" && Preco.CompareTo(other.Preco) > 0)
    {
        result = true;
    }

    // Ordena por preço (maior para menor)
    if (resposta == "4" && Preco.CompareTo(other.Preco) < 0)
    {
        result = true;
    }

    // Ordena por data (mais próximo da data atual)
    if (resposta == "5")
    {
        // pega a data do primeiro evento e subtrai pela data atual
        TimeSpan TempoPrimeiroEvento = Data - DateTime.Now;

        // faz o mesmo com a data do outro evento
        TimeSpan TempoOutroEvento = other.Data - DateTime.Now;

        // depois de fazer essas contas ele faz a verificação para retornar o valor do resultado
        if (TempoPrimeiroEvento > TempoOutroEvento)
        {
            result = true;
        }
    }

    return result;
}
}
```