



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



# Ограничители при шаблонните класове

Templates

# Съдържание

1. Защо са ни нужни?
2. Ограничаване до референтен тип
3. Ограничаване до референтен тип
4. Ограничаване до конструктор
5. Ограничаване до статичен базов клас
6. Ограничаване до шаблонен базов клас
7. Комбиниране на ограничения

# Проблем

- Кодът, генериран за `Equals<string>`, ще е различен от този за `Equals<int>`

```
public static bool Equals<T> (T t1, T t2)
{
    return (t1 == t2);
}
```

- Може да е дори още по-различен, ако сравняваните **типове са предефинирали оператора ==**

# Ограничаване до референтен тип

- Ограничителите се представят в C# с ключовата дума **where**
- Указване, че T трябва да е **референтен тип**

```
public void MyMethod<T>()  
    where T : class  
{  
    ...  
}
```

- **class** тук е ключова дума и трябва да е с малки букви

# Ограничаване до примитивен `min`

- Указване, че `T` трябва да е `примитивен min`

```
public void MyMethod<T>()  
    where T : struct  
{  
    ...  
}
```

- `struct` тук е ключова дума и трябва да е с малки букви

# Ограничаване до конструктор

- Указване, че T трябва да е **конструктор**

```
public void MyMethod<T>()  
    where T : new ()  
{  
    ...  
}
```

- Само **default конструктор** може да бъде използван
- **Параметризиран конструктор** ще доведе до грешка при компилиране

# Ограничаване до статичен базов клас

- Указване на статичен **базов клас** като ограничение

```
public void MyMethod<T>()  
    where T : BaseClass  
{  
    ...  
}
```

- Типът **параметър** трябва да е от **указания** базов клас или да е негов **наследник**

# Ограничаване до шаблонен базов клас

- Указване на **шаблонен базов клас** като ограничение

```
public void MyMethod<T,U>()  
    where T : U  
{  
    ...  
}
```

- Типът **параметър** за **T** трябва да е от **класа-параметър U** или да е негов **наследник**



# Комбиниране на ограничителите

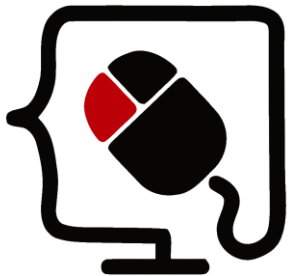
- Указване на **няколко базови класа** и **конструктор** като ограничение

```
public void MyMethod<T>()  
    where T : IComparable, MyBaseClass, new ()  
{  
    ...  
}
```

- Невалидни комбинации от ограничители: **class** и **struct**

# Обобщение

- **Ограничителите** помагат да се ограничи типовия параметър до:
  - референтен тип
  - примитивен тип
  - конструктор
  - статичен базов тип
  - шаблонен базов тип
- Те могат да се **комбинират**



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



Документът е разработен за нуждите на Национална програма "Обучение за ИТ умения и кариера" на Министерството на образованието и науката (МОН) и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).