# Шаблонни (типизирани) класове (templates, generics) – въведение

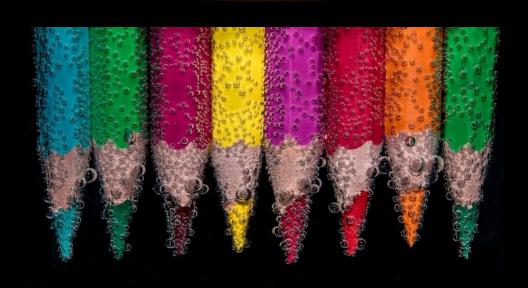


Учителски екип

Обучение за ИТ кариера

https://it-kariera.mon.bg/e-learning/





#### Съдържание

- 1. Защо въобще са ни нужни?
- 2. Синтаксис на шаблонните класове
- 3. Предимства от използването им
- 4. Видимост на параметъра за типа
- 5. Разгъване на шаблонен клас



#### Задача

Нужен ни е клас, който ще съхранява само низове

```
ArrayList strings = new ArrayList();
strings.Add("1");
strings.Add("2");
strings.Add(3); // ОК ли e?!
String e1 = (string) strings.Get(0);
String e2 = (string) strings.Get(2);
String e3 = (String) strings.Get(3); // RTE
```

#### Шаблонни класове – предимства

Трябва ни клас, който ще съхранява само низове

```
IList<string> strings = new List<string>();
strings.Add("1");
strings.Add("2");
strings.Add(3); // грешка при компилация
```

- Добавя проверка на типа (Туре Safety) на клиента
- Осигурява мощен начин за повторно използване на кода

```
IList<int> strings = new List<int>();
IList<Person> people = new List<Person>();
```

#### Шаблонни класове – синтаксис

Дефинира се с <Тип 1, Тип 2 ... и т.н.>

```
class List<T> {
    ...
}
```

Може да има множество параметри за типове

#### Видимост на параметъра за типа

Може да бъде използван навсякъде в класа

```
class List<T> {
    public Add (T element) {...}
    public T Remove () {...}
    public T { get; }
```

#### Задача: кутия с Т

- Създайте клас Вох<>, който може да съхранява всичко
- Добавянето трябва да добавя новото най-отгоре
- Премахването да взима най-горния елемент
- Трябва да има два публични метода:
  - void Add(element)
  - element Remove()

## Решение: кутия с Т

```
public class Box<T>
  //TODO: Добавете полета и конструктор
  public int Count => this.data.Count;
  public void Add(T item)
  { this.data.Add(item); }
  public T Remove()
    var rem = this.data.Last();
    this.data.RemoveAt(this.data.Count - 1);
    return rem;
```

#### Разгъване на шаблонен клас

Можете да го разширите с конкретен клас

```
class BoxOfPickles : Box<Pickle> {
    ...
}
```

```
BoxOfPickles jar = new BoxOfPickles();
jar.Add(new Pickle());
jar.Add(new Vegetable()); // Error
```

## Обобщение

- Шаблонните класове добавят проверка на типа
- Кодът със шаблонни класове е по-лесен за повторна употреба
- Параметърът за типа може да се използва навсякъде в описанието на класа



## Шаблонни класове



https://it-kariera.mon.bg/e-learning/

## Министерство на образованието и науката (МОН)

 Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "Обучение за ИТ кариера" на МОН за подготовка по професия "Приложен програмист"





 Курсът е базиран на учебно съдържание и методика, предоставени от фондация "Софтуерен университет" и се разпространява под свободен лиценз СС-ВҮ-NC-SA



