ДОКЛАД

По

Обектно-ориентирано програмиране

Тема:Въведение в шаблонните класове

1.Шаблонни класове

2.Шаблонни класове и интерфейси

3.Ограничители за шаблонни класове:

Изготвил:Валентин Кьосев

1.Шаблонни класове:

-Шаблонни типове са концепция за типизиране на параметри в програмирането. Те позволяват да се изработват класове и методи, които да отложат спецификацията на един или повече типове данни, докато един клас или метод не се декларира и не получи своята инстанция. Например, чрез използване на шаблонен тип параметър T може да се напише един клас, който друг код също може да използва, без да понесе забавяне на времето на работа.

2.Шаблонни методи и интерфейси:

**А) Какво е „интерфейс“?**

* Тип данни
* Декларира единствено поведение
* Не описва реализация на поведението
* Сходен с напълно абстрактен клас (клас без полета, всички членове на който са абстрактни)
* Може да се разглежда като „договор“ между тип данни, който реализира декларираното поведение, и код, който го използва
* Интерфейсите се имплементират от класове/структури – имплементиращият тип е длъжен да предостави имплементация на съответните членове (може и абстрактна)
* Един клас/структура може да имплементира няколко интерфейса (множествено наследяване?)

**Б) Интерфейси в C#**

Референтни типове данни:

* Не може да бъде създаден екземпляр на интерфейс
* Могат да декларират единствено методи, свойства и индексатори
* Всички членове са с публична видимост
* Всеки интерфейс може да наследява един или няколко други интерфейса; при това той автоматично получава членовете, декларирани в родителските интерфейси
* Интерфейс може да наследява единствено интерфейси
* Прието е да се именуват с представка I (от interface)

3.Ограничители за шаблонни класове:

**А) Ограничаване до референтен тип:**

Ограничителите се представят в C# с ключовата дума where

▪ Указване, че Т трябва да е референтен тип

public void MyMethod< T >()

where T : class

{

...

}

▪ class тук е ключова дума и трябва да е с малки букви

**Б) Ограничаване до конструктор:**

Указване, че Т трябва да е конструктор

▪ Само default конструктор може да бъде използван

▪ Параметризиран конструктор ще доведе до грешка при

Компилиране

public void MyMethod< T >()

where T : new ()

{

...

}

Само default конструктор може да бъде използван

▪ Параметризиран конструктор ще доведе до грешка при

Компилиране

**В) Ограничаване до шаблонен базов клас:**

Указване на шаблонен базов клас като ограничение

▪ Типът параметър за T трябва да е от класа-параметър U или

да е негов наследник

Ограничаване до шаблонен базов клас

public void MyMethod< T, U >()

where T : U

{

...

}

▪ Типът параметър за T трябва да е от класа-параметър U или

да е негов наследник

**Г)Комбиниране на ограничителите:**

▪ Указване на няколко базови класа и конструктор като

ограничение

▪ Невалидни комбинации от ограничители: class и struct

Комбиниране на ограничителите

public void MyMethod< T >()

where T : IComparable, MyBaseClass, new ()

{

...

}

▪ Невалидни комбинации от ограничители: class и struct

**Източници:**

<file:///C:/Users/user/Downloads/03.3.%20%D0%9E%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D0%B8%20%D0%B7%D0%B0%20%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BD%D0%B8%20%D0%BA%D0%BB%D0%B0%D1%81%D0%BE%D0%B2%D0%B5.pdf>

<https://bg.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BD%D0%B8_%D1%82%D0%B8%D0%BF%D0%BE%D0%B2%D0%B5>

**Изготвил:Валентин Кьосев , 11,,а‘‘ клас**