

Комуникацията между обекти. Арументи на събития



Учителски екип

Обучение за ИТ кариера

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



Съдържание

1. Аргументи/данни на събития
2. Наследници на класа EventArgs
3. Дефиниране и закачване на обработчици на събития
4. Анонимни функции
5. Цикъл на събитие



Данни на събитие (EventArgs)

- Класът EventArgs се ползва в сигнатурата на много делегати и обработчици на събития:

```
public delegate void EventHandler  
                    (object sender, EventArgs e);
```

- Когато потребителските данни е необходимо да бъдат предавани, класа EventArgs може да бъде разширен.

```
public event EventHandler<EventArgs> WorkPerformed;
```

Наследници на класа EventArgs

```
public class WorkPerformedEventArgs : EventArgs
{
    public int Hours { get; set; }
    public WorkType WorkType { get; set; }
}
```

- За използване на потребителски клас EventArgs, делегатът трябва да указва в сигнатурата си

```
public delegate void WorkPerfHandler
    (object sender, WorkPerfEventArgs e)
```


Дефиниране и закачване на обработчици на събития

- Операторът `+=` се ползва за закачване на събитие към обработчик на събитие:

```
var worker = new Worker();  
worker.WorkPerformed +=  
new EventHandler<WorkPerfEventArgs>(worker_WorkPerformed);
```

```
var worker = new Worker();  
worker.WorkPerformed += worker_WorkPerformed;
```

Компиляторът
ще **"infer"**
делегата

```
void worker_WorkPerformed(object sender, WorkPerfHandler e)  
{  
    Console.WriteLine(e.);  
}
```

Анонимни методи

- Анонимните методи позволяват кода на обработчика да бъде закачен директно към събитието
- Анонимните методи се дефинират, използвайки ключовата дума **delegate**:

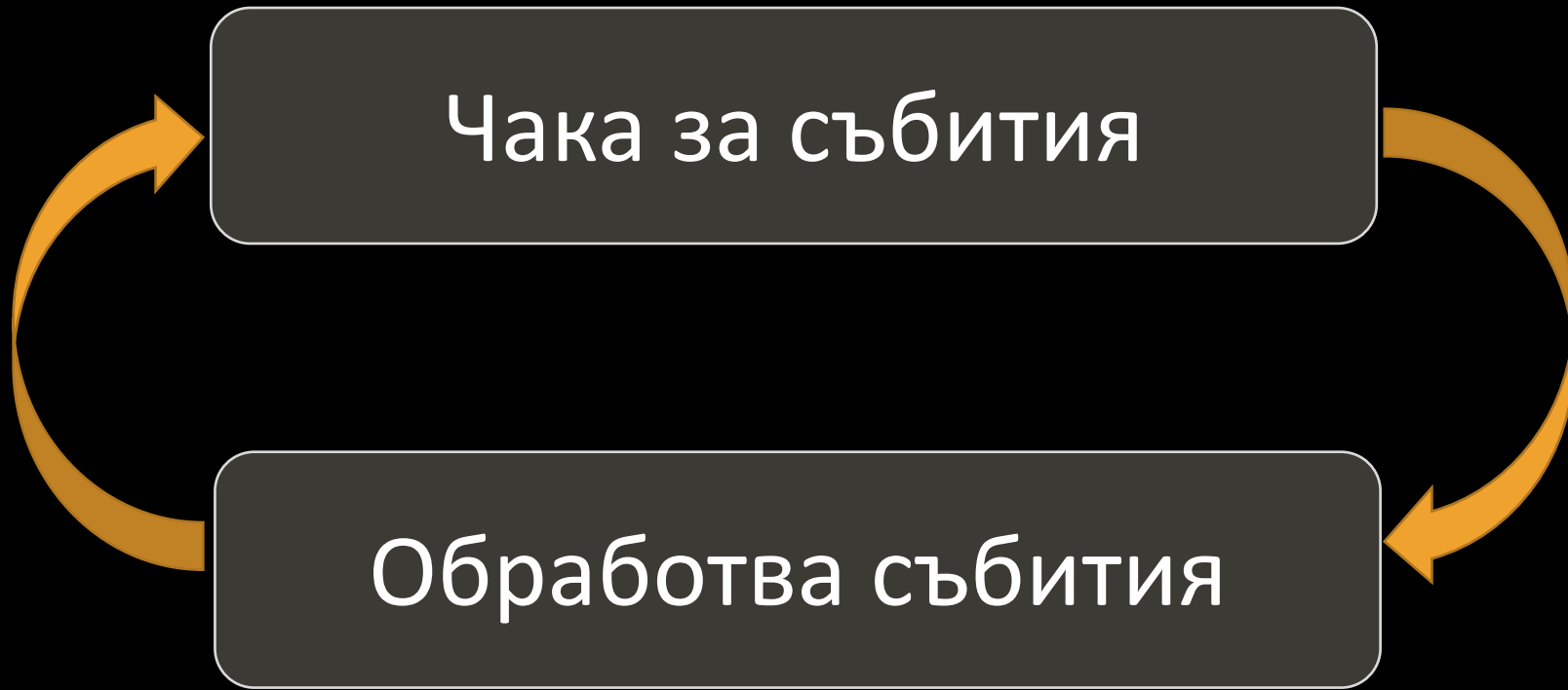
```
var worker = new Worker();  
worker.WorkPerformed +=  
delegate(object sender, WorkPerfEventArgs e)  
{  
    Console.WriteLine(e.Hours.ToString());  
};  
// Край на анонимния метод
```

Обработка на събитието Click на мишката UI – Пример

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        this.InitializeComponent();
        this.MouseDown += this.MainWindow_MouseClick;
    }
    private void MainWindow_MouseClick(object sender,
                                        MouseButtonEventArgs e)
    {
        MessageBox.Show(string.Format("Mouse clicked at ({0}, {1})",
            e.MouseDevice.GetPosition(this).X,
            e.MouseDevice.GetPosition(this).Y));
    }
}
```

Получава данни за
щракване на мишката като
MouseButtonEventArgs

Цикъл на събитие



Цикъл на събития – прост пример

```
while (message != "quit")  
{  
    // Блокиране на операция – чака за събитие на ОС  
    message = GetMessage();  
  
    ProcessMessage(message);  
}
```

Какво научихме?

- Слушателите се записват с += и отписват с -=
 - Когато събитие „се случи“, всички абонат
- Анонимните методи позволяват кода на обработчика да бъде закачен директно към събитието
- Обработката на събитията се извършва циклично



Комуникация между обекти. Аргументи на събития



Въпроси?



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"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

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



SoftUni
Foundation

