

# Комуникация между обекти. Събития



Учителски екип

Обучение за ИТ кариера

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



# Съдържание

1. Събития
2. Дефиниране на събития
3. Известяване за събития
4. Събития и делегати
5. Потребителски събития



# Събития

- В компонентно-ориентираното програмиране компоненти публикуват **събития** към други компоненти
  - Събитията известяват, че нещо се е случило
- Обектът, който предизвиква дадено събитие се нарича **подател на събитие**
- Обектът, който получава събитие се нарича **получател на събитие**
- За да получат едно събитие, получателите на събитието първо трябва да се "**абонират за събитието**"

# Дефиниране на събитие

- Събитията в C# са специални инстанции на **делегати**, декларирани с ключовата дума **event**

```
public event WorkPerformedHandler WorkPerformed;
```

```
public event EventHandler WorkCompleted;
```

- **+=** subscribes for an event
- **-=** unsubscribes for an event



# Известяване за събитие

- Известяване за събитие се прави чрез извикване на събитието като метод:

```
if (WorkPerformed != null)  
    WorkPerformed(8, WorkType.GenerateReports);
```

- Или чрез достъп до делегата на събитето, като се извика директно:

```
WorkPerformedHandler del =  
    WorkPerformed as WorkPerformedHandler;  
if (del != null)  
    del(8, WorkType.GenerateReports);
```

# Събития и делегати

```
public Action<string> m;
```

≠

```
public event Action<string> m;
```

- Събитията могат да са част от интерфейс – делегатите не могат
- Събитие може да се извика единствено в класа в който е дефинирано
- По подразбиране **достъпът** до събитията е синхронизиран (thread-safe)

# Известяване и отговор на събития

```
public delegate void WorkPerfHandler (int hours, WorkType workType);  
public class Worker  
{  
    public event WorkPerfHandler WorkPerformed;  
    public virtual void DoWork(int hours, WorkType workType)  
    {  
        // Свършете работата и уведомете тук, потребителя, че работата е изпълнена  
        OnWorkPerformed(hours, workType);  
    }  
    //TODO: Добавете метод за raising събитие (следващия слайд)  
}
```

## Известяване и отговор на събития (2)

```
protected virtual void OnWorkPerformed
    (int hours, WorkType workType)
{
    WorkPerfHandler del = WorkPerformed as WorkPerfHandler;

    if (del != null) //Слушателите са прикачени/абонирани
    {
        del(hours, workType);
    }
}
```



# Потребителски събития: конвенция

Не връща стойност

Име на делегата: Глагол + EventHandler

```
public delegate void ItemChangedEventHandler(  
    object sender, ItemChangedEventArgs eventArgs);
```

Подателят на събитието е  
`System.Object`

Информацията (данните) за  
събитието наследява  
`System.EventArgs`

## Потребителски събития: конвенция (2)

Събитията се  
декларират като  
**public**

Спазвайте **PascalCase**  
ковенция за  
именуване

```
public event ItemChanged EventHandler ItemChanged;
```

Завършва с **глагол**

## Потребителски събития: конвенция (3)

- За „запалване“ (стартиране) на събитие се създава специален **protected void** метод
  - Именуван от специфичното действие, което върши, напр. **OnVerb()**

```
protected void OnItemChanged()  
{ ... }
```

- Методът получател е именуван във формат **OnObjectEvent**:

```
private void OnOrderListItemChanged()  
{ ... }
```

# Какво научихме?

- Делегатът е **специализиран клас** често наричан “указател към функция”
- **Събитията** позволяват абонамент за известия относно нещо, случващо се в обекта
  - Когато **събитие** „се случи“, всички абонати се известяват





# Комуникация между обекти. 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

