

# 4

## Objektové vlastnosti

Delegáti, udalosti,  
výnimky, Nuget

# Obsah

- Delegáti, udalosti
- Výnimky
- Nuget

# Úloha 1 – sťahovanie a otvorenie súboru

- **Vytvorte konzolovú aplikáciu, pomocou ktorej stiahnete a otvoríte napr. tento súbor:**
  - <http://www.fri.uniza.sk/uploads/files/1444840019-IT-trhovisko15a.pdf>
- **Poznámka:**
  - Na sťahovanie súborov môžete použiť triedu [WebClient](#) a na otvorenie/spustenie súboru triedu [Process](#)

# Úloha 2 – asynchrónne sťahovanie

- Ak ste nepoužili asynchrónne sťahovanie súboru, **upravte program tak, aby ste súbor stiahli pomocou asynchrónnej metódy `DownloadFileAsync`**
- **Zaregistrujte sa na udalosť `DownloadFileCompleted`, v ktorej otvoríte stiahnutý súbor**
- Poznámka:
  - Registrovať udalosť môžeme pomocou jedného z troch spôsobov (lambda výraz, anonymný delegát, klasická pomenovaná metóda) – vyskúšajte si všetky tri spôsoby

# Úloha 3 – vlastné udalosti

- **Stiahnite si projekt z minulého cvičenia a upravte triedu `DoublyLinkedList` tak, že do nej pridajte udalosti:**
  - **Changed**, ktorá sa vyvolá pri každej zmene obsahu (t. j. po pridaní alebo odstránení prvku zoznamu)
  - **ItemAdded**, ktorá sa vyvolá pri pridaní nového prvku do zoznamu
  - **ItemRemoved**, ktorá sa vyvolá pri odstránení prvku zo zoznamu
- **Poznámka:**
  - Udalosť môžete vytvoriť buď prostredníctvom vstavaných delegátov `EventHandler` a `EventHandler<T>` alebo pomocou vlastného definovaného delegáta – vyskúšajte všetky možnosti

# Úloha 4 – vlastné udalosti

- Otestujte všetky udalosti v aplikácii, napr.:

```
var list = new DoublyLinkedList<string>();  
list.Changed += ListOnChanged;  
list.Add("jablko");
```

```
private static void ListOnChanged(object sender, EventArgs eventArgs)  
{  
    Console.WriteLine("List was changed.");  
}
```

# Úloha 5 – výnimky, logovanie

- Práca s výnimkami
  - Vyvolanie výnimky:
    - throw;
    - throw výnimka\_z\_catchu; - pozor, resetuje to zásobník!
    - throw new Exception();
  - Zachytenie výnimky
    - Try { } catch { }
- Logovanie chýb (Debug / Trace triedy alebo cez externé logovacie knižnice – Log4Net, NLog, ...)

# Úloha 5 – výnimky, logovanie

- Pridajte do projektu knižnicu NLog cez NuGet a konfiguračný súbor NLog.config
- Logujte:

```
private static Logger logger =  
    LogManager.GetCurrentClassLogger();  
  
...  
logger.Info("Sample informational message");  
logger.Error("Sample error message");
```



# Úloha 6 – NLog.config

- Upravte konfiguračný súbor takto:

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <!-- See http://nlog-project.org/wiki/Configuration_file
  for information on customizing logging rules and outputs. -->
  <targets>
    <target name="file" xsi:type="File" fileName="${basedir}/logs/${shortdate}.log"
            layout="${longdate} ${uppercase:${level}} ${message} ${exception:format=tostring}" />
    <target name="debugger" xsi:type="Debugger"
            layout="${longdate} ${uppercase:${level}} ${message} ${exception:format=tostring}" />
  </targets>

  <rules>
    <logger name="*" minlevel="Trace" writeTo="file" />
    <logger name="*" minlevel="Trace" writeTo="debugger" />
  </rules>
</nlog>
```