

## 22. Uživatelské rozhraní - Události, Kontrola vstupů, Layout

### Uživatelské rozhraní (User Interface)

Slouží pro interakci mezi uživatelem a programem. Mělo by být uživatelsky přehledné a funkční  
Existuje několik druhů uživatelských rozhraní.

### CLI (Command Line Interface)

Příkazový řádek ve kterém uživatel komunikuje s programem, psaním příkazů. Nevyužívá myš ani menu a nedovede pracovat s celou plochou obrazovky (full screen).  
Příkazový řádek může být spuštěn v textovém terminálu nebo pomocí emulace terminálu, jako je např. PuTTY.  
Spuštěný program může na terminál vypsát hlášení o svém běhu, výsledek operace, vyžádat si další vstup.

### Windows Forms (.NET Framework)

Kniha tříd pro tvorbu GUI (Graphical User Interface)  
Obsahuje předpřipravené tools, jako button, label, DataGridView, ListView, CheckBox, ...  
Vývojové prostředí obsahuje designer, který umožní graficky designovat aplikaci.  
Designer píše zdrojový kód třídy Designer.cs

### Windows Presentation Foundation (.NET Framework)

Nástupce Windows Forms, je novější, celkově se v něm dá udělat mnohem lepší a modernější design. Obsahuje u prvků více vlastností.  
Data o designu prvků jsou v jazyce XAML  
Má lepší designer, mnohem pečlivější a přesnější úpravy prvků  
Jeho cílem je sjednotit uživatelské rozhraní, 2D a 3D grafiku, vektorovou a rastrovou grafiku, animace a provázat s daty audio a videa.  
Funguje pomocí Direct3D knihoven.

Dalo by se říct, že Windows Forms je pro vývoj aplikace, která se nebude dále vylepšovat. Windows Presentation Foundation je pro aplikace, kde záleží na designu a aplikaci, které se plánují vylepšovat. Důvod: Ve Forms se pravděpodobně při vývoji nalezne nejlepší možné řešení. Ve Foundation je řešení spousta a dají se lépe kombinovat, což umožní vylepšování.

## UWP (Universal Windows Platform)

je podobný WPF, často se používá pro vývoj tabletových/mobilních aplikací s windows.

## Události (Event)

Událost je jako delegát, který je odkaz na metodu.

Je to kolekce, do které se ukládá více delegátů.

Jakmile událost nastane, spustí se všechny metody, které jsou v ní uloženy. Nejlehčí případ události je třeba button, když na něj 2krát kliknu, tak se vygeneruje metoda

Sender je objekt, který událost vyvolal. E jsou dodatečné parametry události.

Na metodu ukazuje delegát události, který byl přidán do kolekce události.

Eventu je spousta, v designeru se najdou pod ikonou blesku.

Třeba využití události je Drag and Drop

Událost DragEnter je aktivována jakmile kurzor na objekt, na kterém událost je něco přetáhne.

V tu chvíli budou v e.Data informace jako třeba file path souborů, které byly přetáhnuty.

Poté se pustí levé tlačítko myši a bude provedena událost DragEnter, kde bude provedena akce se soubory

## Kontrola vstupů

Provádí se, aby uživatel nemohl primitivní chybou zapříčinit špatné fungování aplikace.

Třeba do kolonky pro tel číslo napsat písmena a program by se poté ze string pokusil udělat int, což by skončilo errorem.

Kontrola vstupů se dá provádět mnoha způsoby: Contains metody, Regulární výrazy, Eventy na textboxu pro stisknutí jakéhokoliv tlačítka a if statement, jestli tlačítko může být zapsáno.

Místo TextBoxu se dá použít MaskedTextBox, ten může mít nastavenou masku (C# nabízí i předpřipravené) nebo vlastní pro zadání pouze čísel. Zároveň nastaví formát jako 111-111-111.

Kontrola vstupů může probíhat i vícekrát.

Příklad

Server s client windows forms aplikací, zadávání příkazů

- MaskedTextBox zjistí jestli text příkazu nemá čísla
- Tlačítko pro provedení zkontroluje regexy jestli příkaz neobsahuje mezery, popřípadě mezery nechá smazat a příkaz odešle
- Server přijetí příkaz převede na lower case.
- Command pattern na serveru pomocí metody Contains v Dictionary zjistí, jestli příkaz má mezi svými příkazy. Pokud ano, tak ho provede.

# Layout

Je to grafické rozložení prvků v aplikaci.

Layout zahrnuje: velikost a poměr stran aplikace,

Může třeba rozdělit GUI aplikace na 3 sloupce, první sloupec se může rozdělit na řádky pro umístění tlačítek.

O layout se většinou stará designer, ten vývojáři předá představovaný vzhled a ten ho vytvoří.

## Ovládací prvky

Umožňují ovládání aplikace

vývojář by se měl snažit, aby ovládání aplikace bylo co možná nejlehčí.

Rozděluje se na

- Fyzické – klávesnice, myš
- Virtuální – v aplikaci: button, menu, všechno v toolboxu

Často má více ovládacích prvků stejnou funkci.

Třeba mazání souborů: označí se soubor a ten může být smazán button "Delete" nebo tlačítkem delete na klávesnici.