JAK PROGRAM ZÍSKÁVÁ UŽIVATELSKÝ VSTUP

- čte vstupní data (z konzole nebo ze souboru/-ů)
- čte i pokyny, co má dělat
- grafické uživatelské rozhraní (GUI)
- API (POSIX, Win32, REST, SQL…)

Historická odbočka: PARC

- 1969, 3000mil od ústředí XEROX
- laserová tiskárna
- myš
- GUI
- desktop
- WYSIWYG editor
- Interpress (předchůdce Postcriptu)
- Ethernet
- OOP
- Smalltalk
- MVC (architektura Model-View-Controller)

- ...

PROGRAMOVÁNÍ ŘÍZENÉ UDÁLOSTMI

Programování řízené událostmi

Co známe z diskrétní simulace:

Můžeme mít složitý systém, který

- nemá nějaké ústřední velení, které by všechno řídilo
- místo toho se skládá z mnoha částí, které jsou připravené na to, aby zpracovávaly události
- ta složitá funkčnost vzniká z toho, že každý dělá to, co umí.

Programování řízené událostmi

Ten hlavní program:

```
while (ještě_není_konec)
{
  Udalost u = DalsiUdalost();
  AktualniCas = u.kdy;
  ZpracujUdalost( u ) ;
}
```

Kdo zpracovává události?

Všichni

Hierarchie objektů

Úřad

Polymorfie a virtuální metody

Události

Od uživatele:

- klávesnice
  - stisknout
  - pustit
- myš
  - stisknout
  - pustit
  - pohnout

Jak je to třeba ve Windows

Komponenty
Hierarchie
Propadání událostí (úřad)
Polymorfismus (virtuální metody)

Transformace lo-level událostí na hi-level

Jak je to v C# a ve VisualStudiu

Knihovna s komponentami
(WFC), WinForms, (VCL), WPF, XNA, WinUI... GTK, Tcl/Tk

Vytvořit aplikaci
Skoro vždy jde spustit
Visuální návrh
Designer-Toolbox-Inspektor
Properties Left, Top, Width, Height, Visible, Enabled, Text
Události
Rozdělený zdrojový kód partial class
Sdílení obslužných proměnných parametr Sender