

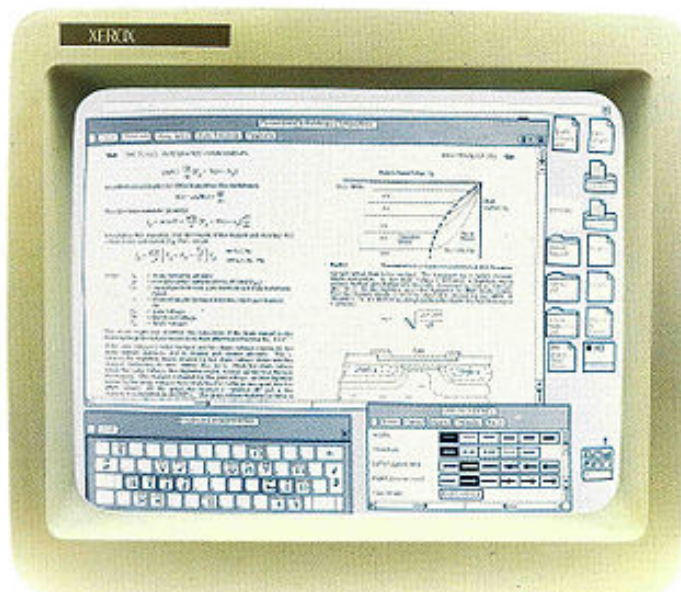
IB111

# Programování a algoritmizace

Graphical User Interface (GUI)

# Historie GUI

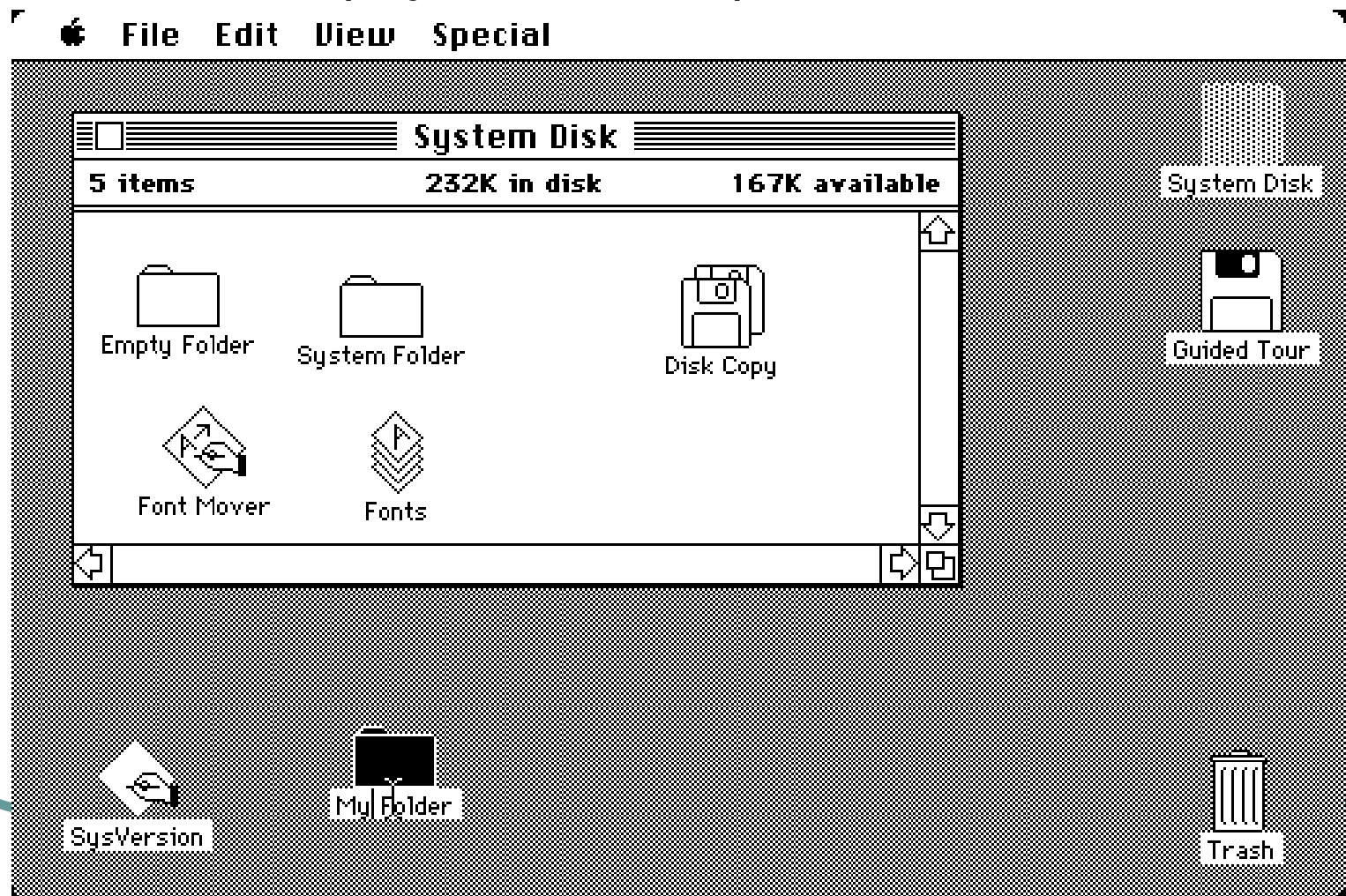
- První OS s GUI: Xerox PARC



- Apple, Microsoft, Sun následovali

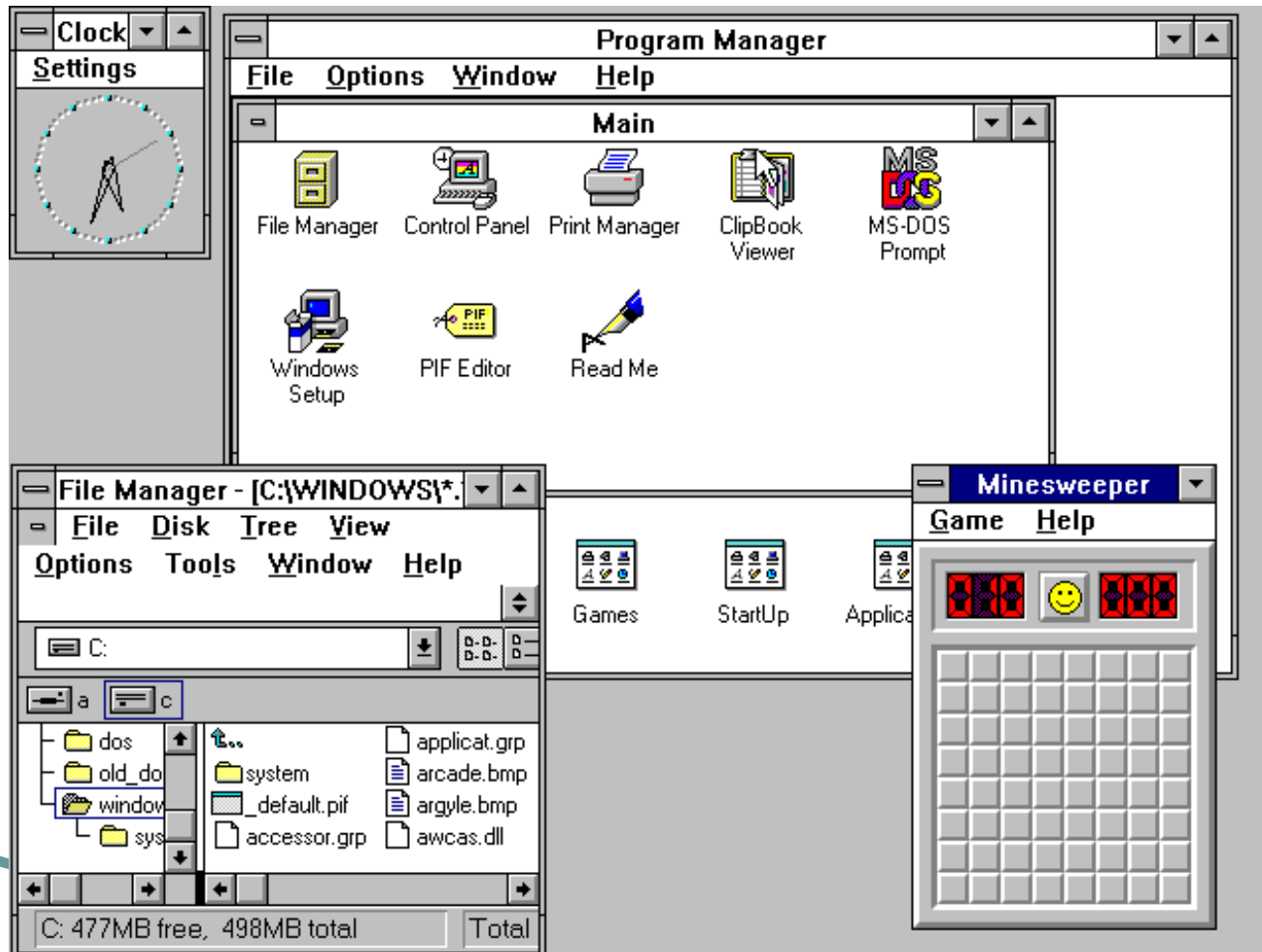
# Historie GUI

- MAC OS (System 1.1) - 1984



# Historie GUI

- Microsoft Windows 3.1 (1992)





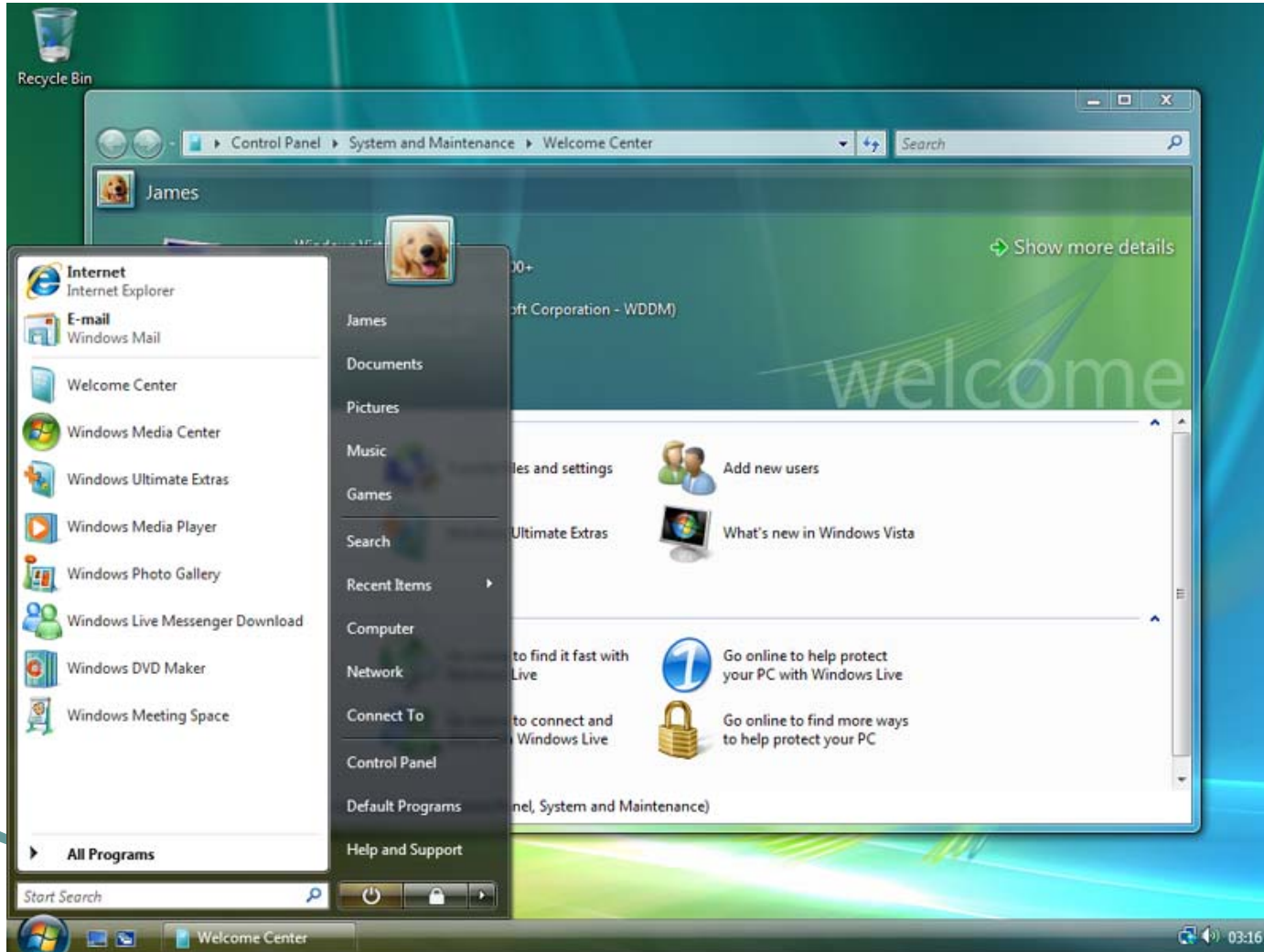
# Historie GUI

- Windows 95



# Historie GUI

- Windows Aero (3D prvky)



# Programování GUI

- Microsoft Windows
  - Programátorské rozhraní Win32
    - Dnes nazývané **Windows API**
    - Několik tisíc funkcí
  - Podpora pro vytváření menu, ikon, bitmap, dialogů apod.
  - Založeno na událostmi řízené architektuře programu
    - Event driven programming



# Programování GUI

- Program pro MS Windows pracuje v následujících krocích
  - registrace třídy okna
  - vytvoření hlavního okna aplikace
  - provádění cyklu, který očekává příchod události
    - jakmile událost přijmeme následuje její předání obslužné funkci okna a zpracování události
    - cyklus končí s příchodem zprávy pro ukončení celé aplikace



# Příklad v C/C++ pro MS Windows

## ● Registrace třídy okna

```
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEX wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc    = (WNDPROC) WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance      = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, (LPCTSTR)IDI_AAA);
    wcex.hCursor        = LoadCursor(NULL, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH) (COLOR_WINDOW+1);
    wcex.lpszMenuName   = (LPCTSTR)IDC_AAA;
    wcex.lpszClassName  = szWindowClass;
    wcex.hIconSm        = LoadIcon(wcex.hInstance, (LPCTSTR)IDI_SMALL);

    return RegisterClassEx(&wcex);
}
```

# Programování GUI

- Vytvoření hlavního okna aplikace

```
☐ BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    HWND hWnd;

    hInst = hInstance; // Store instance handle in our global variable

    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

☐ if (!hWnd)
    {
        return FALSE;
    }

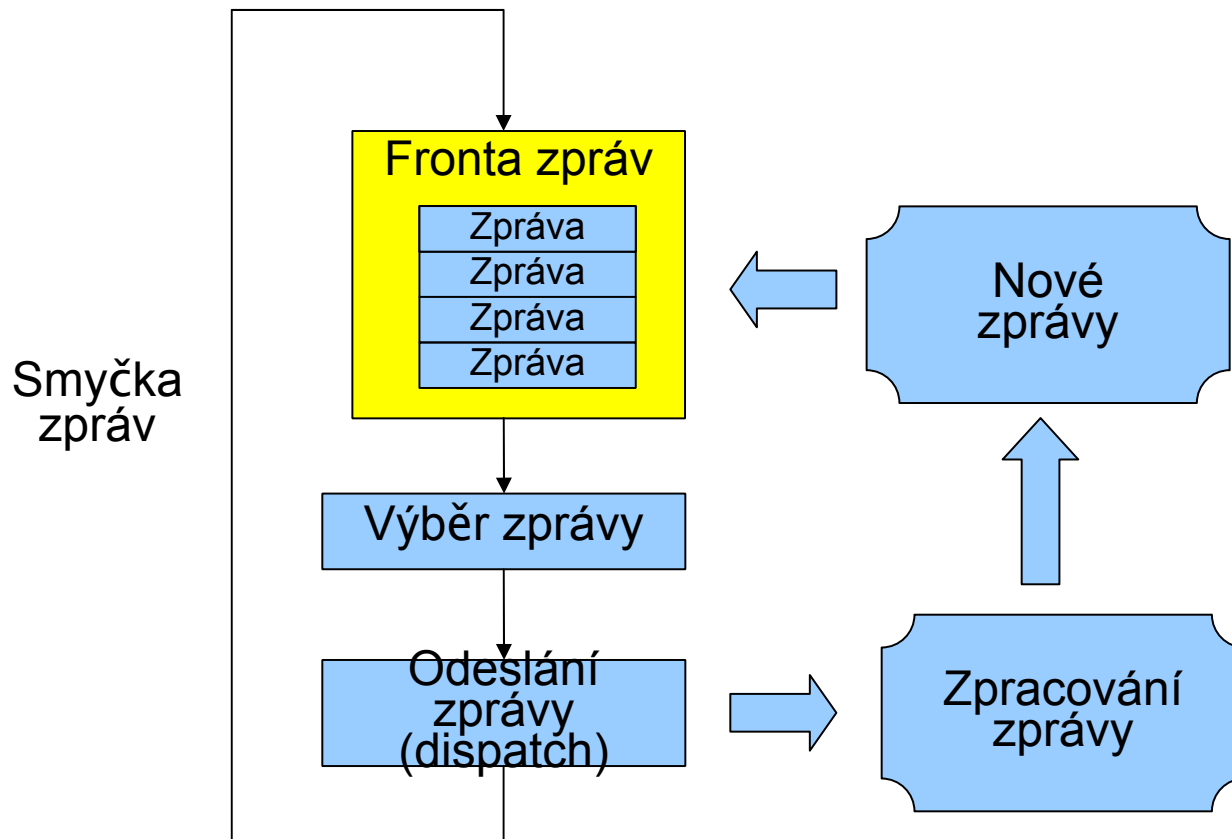
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}
```

# Programování GUI

- Událostmi řízené programování
  - Vstup od uživatele znamená vytvoření události
    - Stisky kláves
    - Přesun/kliknutí myši
  - Program reaguje na jednotlivé události a zpracovává (ošetřuje) je.

# Událostmi řízené programování



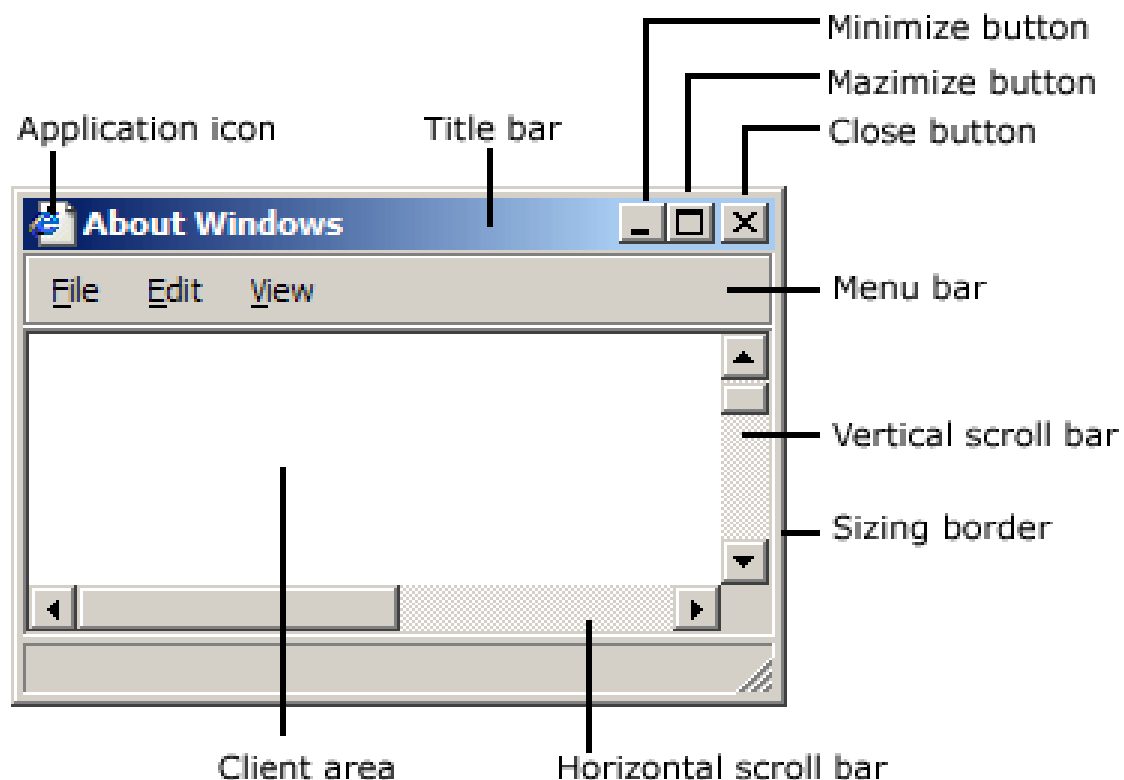
# Příklad v C/C++ pro MS Windows

```
▣ LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    PAINTSTRUCT ps;
    HDC hdc;

    ▣ switch (message)
    {
        ▣ case WM_COMMAND:
            wmId    = LOWORD(wParam);
            wmEvent = HIWORD(wParam);
            // Parse the menu selections:
            switch (wmId)
            {
                case IDM_ABOUT:
                    DialogBox(hInst, (LPCTSTR)IDD_ABOUTBOX, hWnd, (DLGPROC)About);
                    break;
                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;
                default:
                    return DefWindowProc(hWnd, message, wParam, lParam);
            }
            break;
        case WM_PAINT:
            hdc = BeginPaint(hWnd, &ps);
            // TODO: Add any drawing code here...
            EndPaint(hWnd, &ps);
            break;
        case WM_DESTROY:
            PostQuitMessage(0);
            break;
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}
```

# Okno

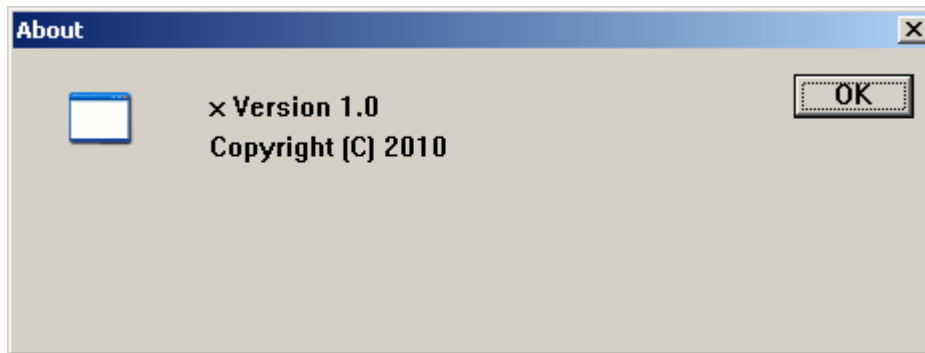
- Základní prvky okna





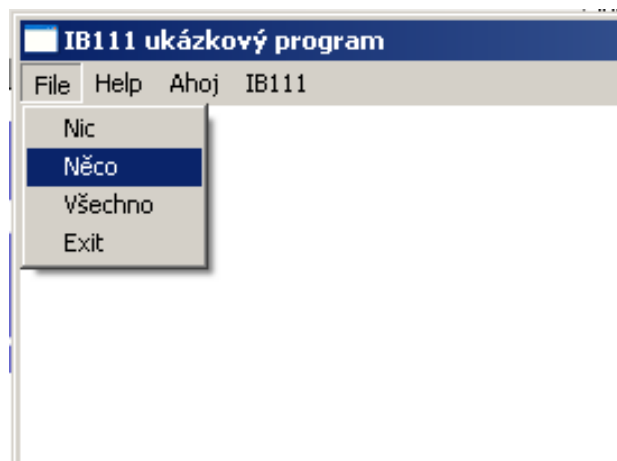
# Okna

- Okno rodičovské a dětské
- Okno rodičovské se často skládá z několika oken dětských
  - Widget – grafické objekt s určitou funkcí
  - Např. tlačítko

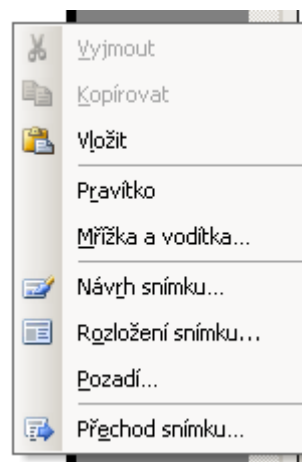


# Prvky GUI

- Hlavní menu



- Kontextové menu



# Prvky GUI

- Button



- Edit Box

A screenshot of a GUI form element. It contains two input fields. The first field is labeled "Jméno:" and the second field is labeled "Příjmení:". Both fields are empty and have a light gray background.

# Prvky GUI

- Check Box

Souhlasím se vším ...

- Radio Buttons

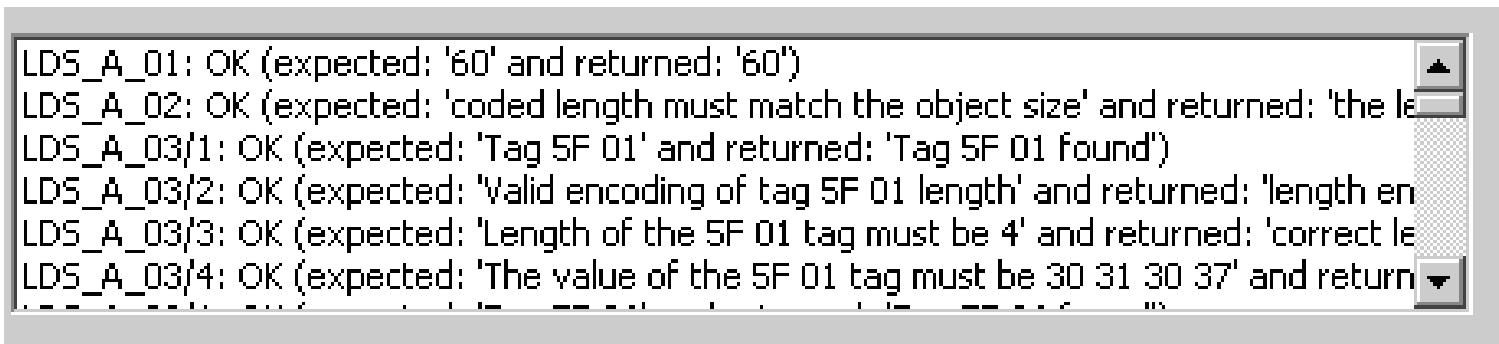
Nechci nic

Chci jen něco

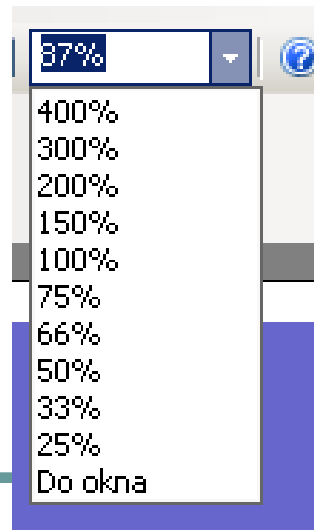
Chci všechno

# Prvky GUI

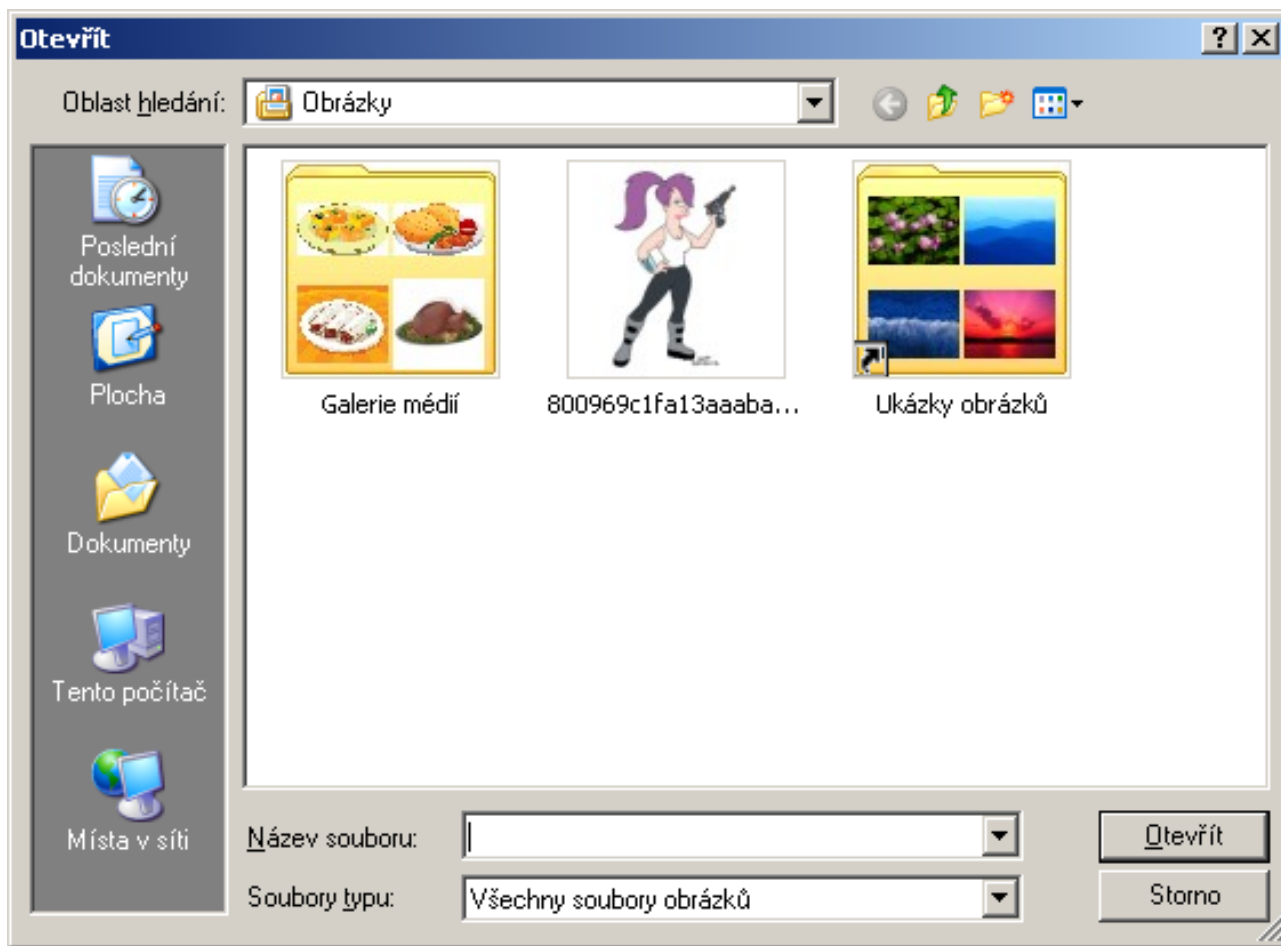
- List Box



- Combo Box



# Společné dialogové boxy



# Historický příklad MacOS

## Ovládací prvky Apple (Finder)

textové políčko

roletové menu ▼

šoupátko

tlačítka spouštějící akci

zaškrtnutá políčka

- |                                    |                                    |
|------------------------------------|------------------------------------|
| <input type="radio"/> Radio Button | <input type="checkbox"/> Check Box |
| <input type="radio"/> Radio Button | <input type="checkbox"/> Check Box |
| <input type="radio"/> Radio Button | <input type="checkbox"/> Check Box |

oblast ovládaná posuvníky

# Historický příklad Windows 3.x

## Ovládací prvky MS Windows 3.x

### textové políčko

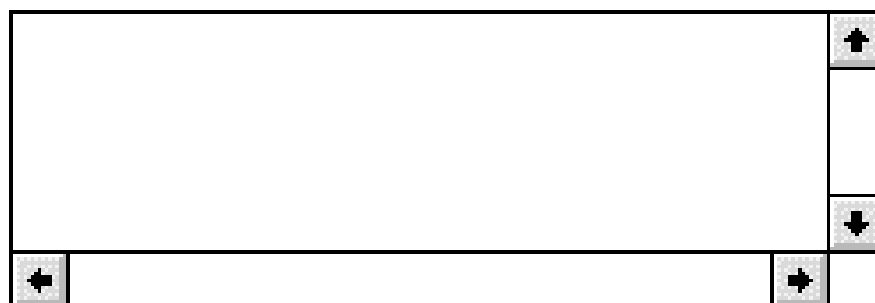
### šoupátko

**tlačítko spouštějící akci**

### zaškrťovací políčka

- |                                    |                                    |
|------------------------------------|------------------------------------|
| <input type="radio"/> Radio Button | <input type="checkbox"/> Check Box |
| <input type="radio"/> Radio Button | <input type="checkbox"/> Check Box |
| <input type="radio"/> Radio Button | <input type="checkbox"/> Check Box |

### oblast ovládaná posuvníky





# GUI v pythonu

- Python má i nemá GUI (jak se to vezme)
- K dispozici na řadě platforem (OS)
  - Tkinter
  - wxWidgets
  - Qt
  - Gtk+
  - FLTK
  - FOX
  - OpenGL
- Plus několik specifických
  - Mac: The Mac port
  - Windows: Pythonwin (používá MFC)

# Tkinter

- Na cvičení si ukážeme několik příkladů v Tkinter
- Začneme importem
  - `import Tkinter`  
nebo
  - `from Tkinter import *`

# Tkinter „Hello world“

```
from Tkinter import *  
root = Tk()  
w = Label(root, text="Hello, world!")  
w.pack()  
root.mainloop()
```



# Základní „widgets“ v Tkinter

Widget	Description
Button	A simple button, used to execute a command or other operation.
Canvas	Structured graphics. This widget can be used to draw graphs and plots, create graphics editors, and to implement custom widgets.
Checkbutton	Represents a variable that can have two distinct values. Clicking the button toggles between the values.
Entry	A text entry field.
Frame	A container widget. The frame can have a border and a background, and is used to group other widgets when creating an application or dialog layout.
Label	Displays a text or an image.
Listbox	Displays a list of alternatives. The listbox can be configured to get radiobutton or checklist behavior.
Menu	A menu pane. Used to implement pulldown and popup menus.
Menubutton	A menubutton. Used to implement pulldown menus.
Message	Display a text. Similar to the label widget, but can automatically wrap text to a given width or aspect ratio.
Radiobutton	Represents one value of a variable that can have one of many values. Clicking the button sets the variable to that value, and clears all other radiobuttons associated with the same variable.
Scale	Allows you to set a numerical value by dragging a “slider”.
Scrollbar	Standard scrollbars for use with canvas, entry, listbox, and text widgets.
Text	Formatted text display. Allows you to display and edit text with various styles and attributes. Also supports embedded images and windows.
Toplevel	A container widget displayed as a separate, top-level window.

# Poslední přednáška



- Následuje cvičení v A104 v 12:00
- Toto byla poslední přednáška v tomto semestru, termíny ZK jsou v ISu, minimálně 75 % bodů z domácích úkolů