Tvorba uživatelských rozhraní

Prezentace přednášek

Ústav počítačové grafiky a multimédií

Téma přednášky

Úvod, tvorba rozhraní, základní principy

Motto

„Proč je tak důležité, aby komunikace člověka se strojem (počítačem) byla efektivní a jak se dnes rozhraní tvoří?“

Obsah:

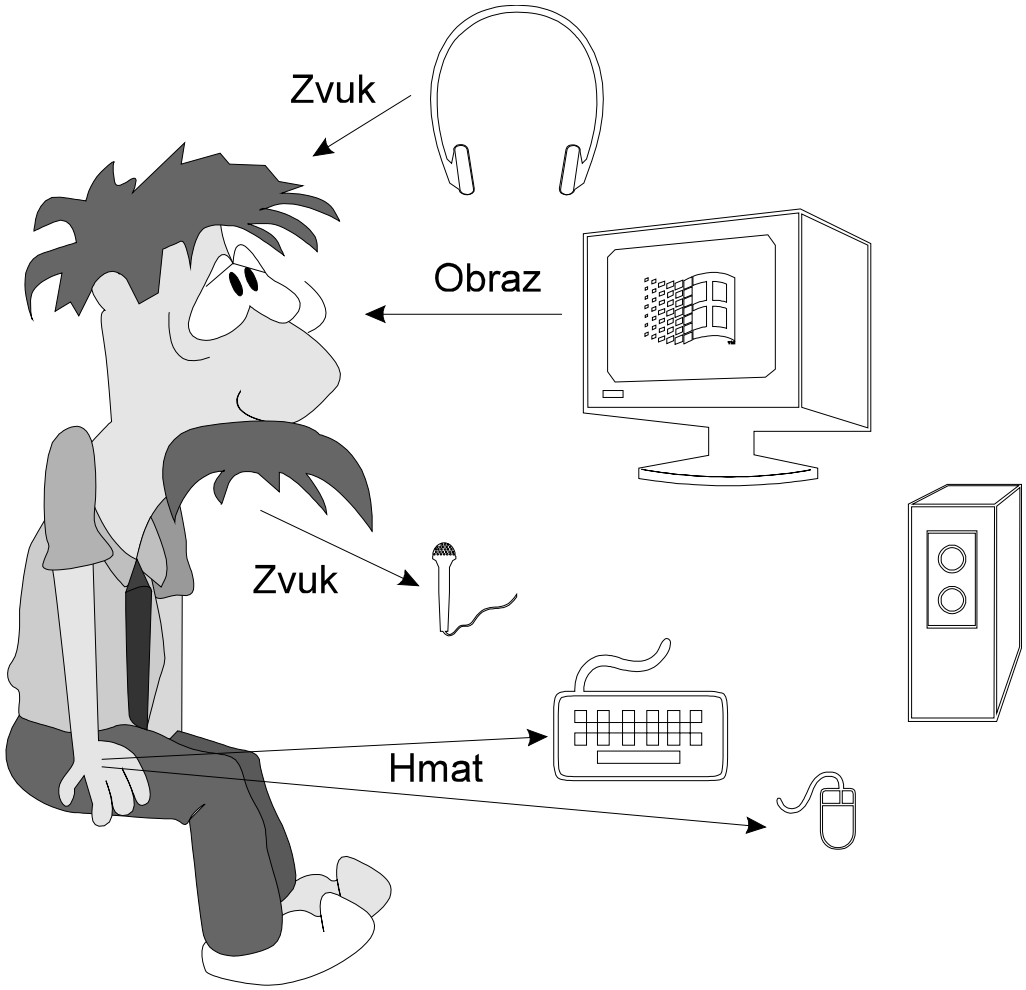
- Úvod
- Komunikační kanály, periferie
- Řízení událostmi
- Objekty a API
- Principy vývojových nástrojů

Úvod

Komunikace člověka s počítačem patří v současnosti mezi nejdůležitější úlohy výzkumu v oblasti počítačů, a to z následujících důvodů:

- Komunikace člověka se strojem značně ovlivňuje efektivitu používání počítačů v nejrůznějších odvětvích lidské činnosti
- Komunikace člověka s počítačem je jednou z nejméně propracovaných oblastí procesu „práce s počítačem“
- Komunikace člověka s počítačem je současně nejužším místem při provádění většiny „běžných činností“ při práci s počítačem
- V komunikaci se strojem je velký potenciál pro zlepšení

Komunikační kanály



Komunikační kanály

- Obraz (zrak) - Vysoká propustnost, "náhodný přístup", jen směr počítač->člověk
- Zvuk (sluch) - Nižší propustnost, "sériový přístup", neodvádí pozornost zraku, komunikace oběma směry
- Hmat - V současné době až na výjimky používaný pouze pro komunikaci člověk->počítač (klávesnicce, myš, ...)
- Čich, chuť - v současnosti pro komunikaci nepoužitelné

Z hlediska styku člověka se strojem jsou zajímavé takové komunikační kanály, které jsou schopny reprodukovatelně a pokud možno nezkresleně přenášet informace, důležitá je možnost digitalizace.

Periferní zařízení - obraz

- Tiskárny - tradiční zařízení, pouze pořizují trvalé (papírové) kopie dokumentů z počítače
- Displeje - z této třídy zařízení jsou zajímavé obrazovky, případně LCD panely; zařízení pro "statický" výstp obrazu
- Videosystémy - (still video) zařízení, která jsou schopna pracovat takovou rychlostí, že zobrazí nejméně 6 (?) obrazů za sekundu v dostatečném rozlišení – typicky PC
- Systémy virtuální reality - jsou taková zařízení, která jsou schopna generovat několik obrazů za sekundu z popisu scény a zobrazit je – většina moderních PC

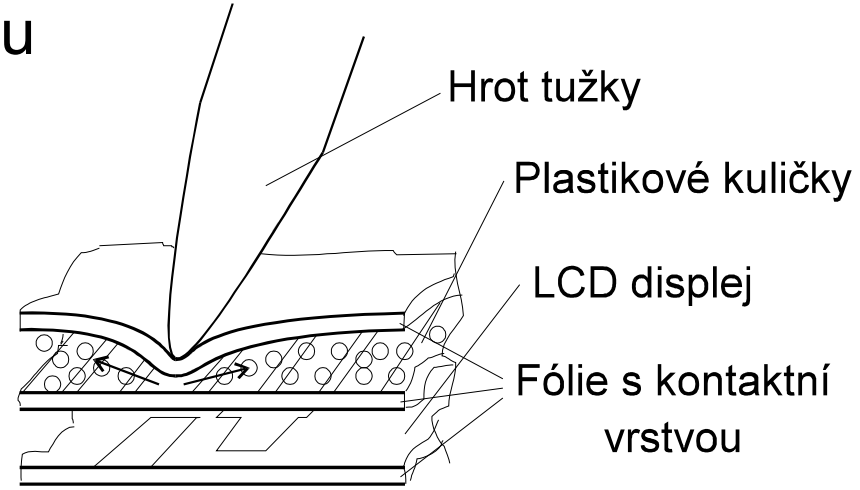
Do této skupiny lze zařadit i systémy pro pořizování a zpracování obrazu, které mohou sloužit pro zpracování kreseb a písma psaného rukou, případně přímo ke zpracování obrazu osoby.

Periferní zařízení - zvuk

- Zařízení pro záznam zvuku - většina číslicových zařízení zaznamenává zvuk pomocí vzorkování; vyskytují se systémy s nejrůznější kvalitou
- Zařízení pro reprodukci zvuku - reprodukují zvuk sejmutý zařízeními z předchozí skupiny
- Zařízení pro rozpoznávání zvuku (řeči) - jedná se o zařízení vyšší kategorie, která jsou schopna rozpoznat význam zvuku a zejména lidské řeči
- Zařízení pro generování zvuku (hudba, řeč) - systémy této třídy jsou schopny generovat zvuk potřebných vlastností; existuje řada principů,na nichž mohou být taková zařízení založena

Periferní zařízení - hmat

- Klávesnice - pravděpodobně dosud nejobvyklejší zařízení pro předávání informace člověkem do počítače
- Myš – notoricky známé zařízení pro grafickou komunikaci v plošných (2D) souřadnicích
- Zařízení pro záznam kresby a rozpoznávání - tato zařízení v současné době postupně nabývají na významu (pen computing); jsou výhodná pro své malé rozměry a jednoduchost ovládání; stávají se alternativou klávesnice i myši, která je výhodná zejména pro rozměry zařízení; často jsou kombinována s displejem



Výhled do budoucna

- „Osobní rádce“ - počítač v kapse, sluchátko a mikrofon, počítač člověku radí, odpovídá mu na otázky... Promítá mu obraz přímo do oka? Čte jeho výraz a gesta?
- „Zabudovaný počítač“ - počítač v těle, napojení na nervový systém... Možná by šlo, ale sneseme to?
- „Integrovaný přístroj“ - komunikace, inteligence a záznam v jednom balíčku... Standardizace?
- „Helma pro virtuální realitu“ - v současnosti stagnace – problémy jsem zejména v praktickém užití (jak si s helmou psát poznámky?) a technických problémech (rozlišení displejů, zpoždění vyvolávající nevolnost při práci s helmou)
- „Rukavice pro snímání polohy ruky“ - obsahuje čidla pro snímání polohy prstů a ruky jako celku; užití ve speciálních aplikacích (simulace) a ve hrách

Současný pohled na rozhraní

Programovací jazyky - jak jimi popíšeme rozhraní?

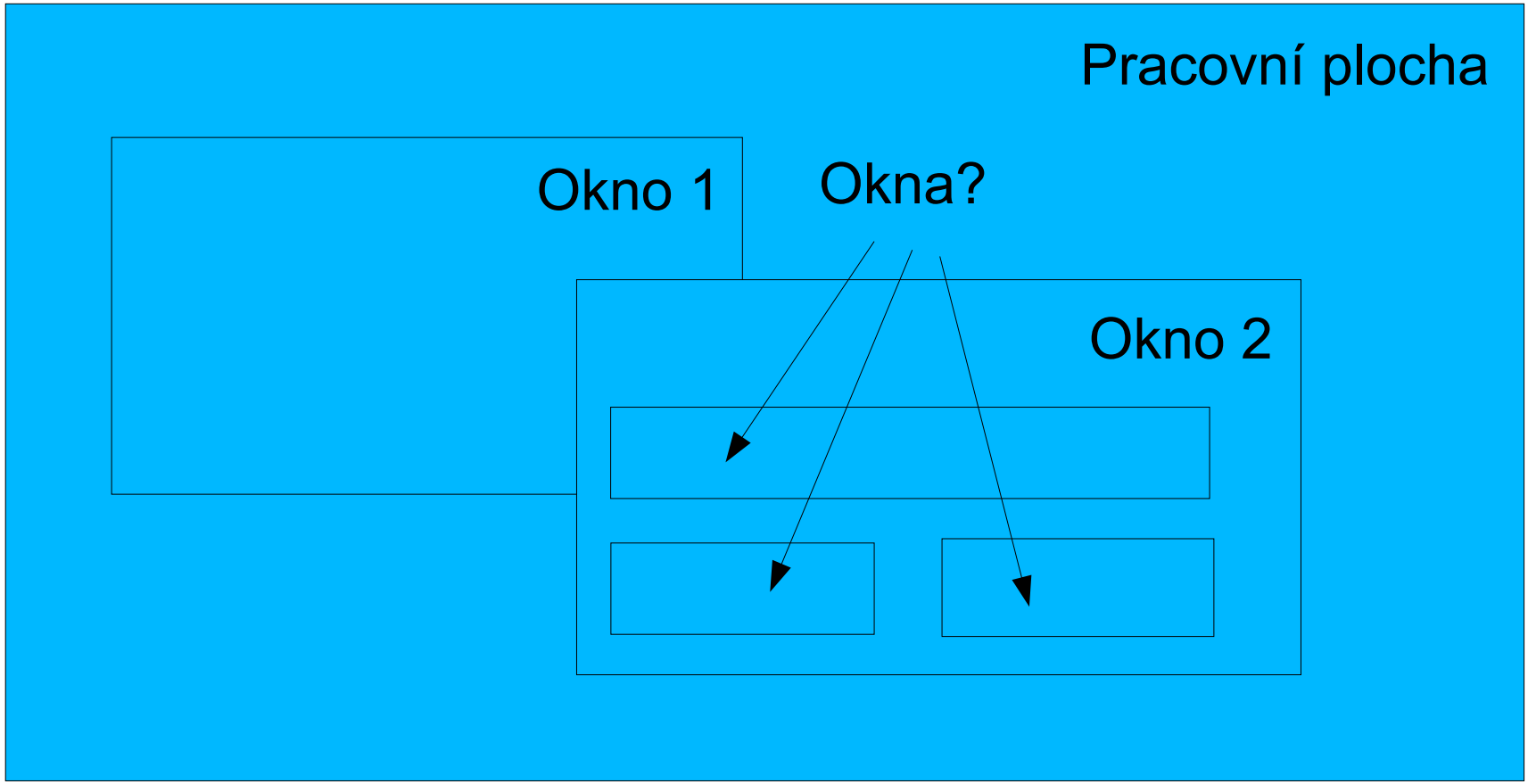
- Textově „přímo“ - tradiční cesta a velmi neefektivní cesta
- Textově „rámcově“ - užívá se, ale dosti výjimečně (Tcl/Tk)
- WYSIWYG – různé metody užívající „prefabrikované“ stavební prvky rozhraní

Platformy

- Grafické prostředí Windows API
- X-Window (UNIXy)
- Mobilní (Symbian, Windows CE...)

Okna v uživatelských rozhraních

Koncept oken pochází od firmy Xerox z 70. let 20. století



Programování rozhraní

Programovací jazyky - jak jimi popíšeme rozhraní?

- Textově „přímo“ - tradiční cesta a velmi neefektivní cesta
- Textově „rámcově“ - užívá se, ale dosti výjimečně (Tcl/Tk)
- WYSIWYG – s použitím prefabrikovaných stavebních prvků různého typu (API s vestavnými funkcemi, komponenty atd.)

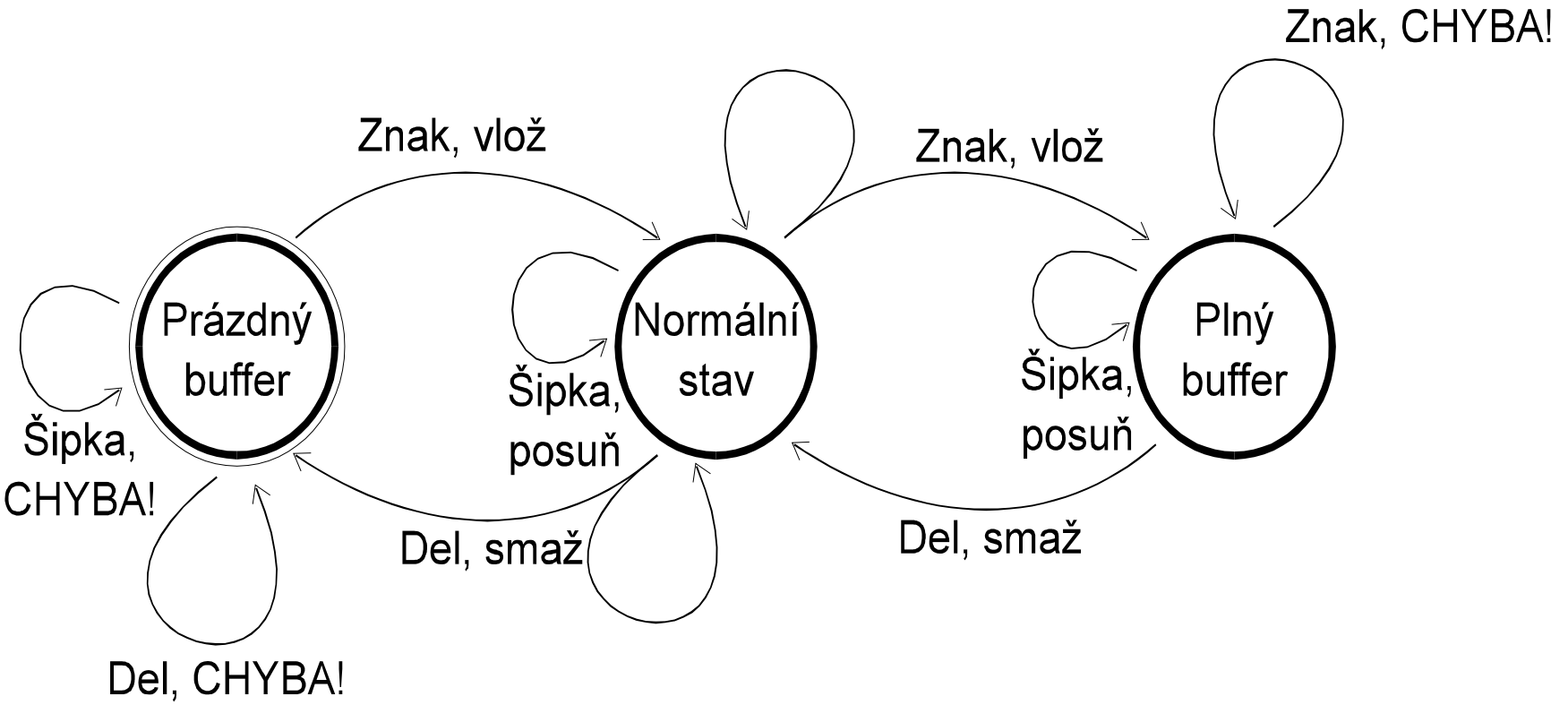
Jaký programovací jazyk?

- Rozšířené a známé Delphi – Pascal
- Nové C# od Microsoft
- C++ (C++ Builder, Visual C++)

Událostmi řízené systémy

Příklad – zadávání textu do „editačního boxu“ - Automat

- Automat je třeba implementovat programem v počítači



Událostmi řízené systémy

Implementace
„Místem v programu“

- Jednoduché a „přirozené“
- Těžko modifikovatelné pro další klávesy (musí se přidat do dlouhého textu)
- Těžko rozšiřitelné (více vstupních řádků)

```
...
Empty:
     for (;;) switch (C = getch()) {
     case Arrow:
          Error();
          break;
     case Del:
          Error();
          break;
     default:
          Insert(C);
          goto Normal;
          }
Normal:
     for (;;) switch (C = getch()) {
     case Arrow:
          Move(C);
          break;
     case Del:
          Delete();
          if (Empty()) goto Empty;
          break;
     default:
          Insert(C);
          if (Full()) goto Full;
          break;
          }
...
```

Událostmi řízené systémy

Implementace
„Stavovou proměnnou“

- Obtížnější (automat je explicitně součástí návrhu)
- Lépe modifikovatelné pro další klávesy (dlouhý text, ale přehledné)
- Lehce rozšiřitelné na více vstupních řádků – vytvoří se jen více instancí dat

```
int HandleKey(char C,int & State,Data * D) {
     switch (State) {
     case Empty: switch (C) {
          case Arrow:
               return (Error);
          case Del:
               return (Error);
          default:
               Insert(C,D);
               State = Normal;
               return(OK);
               }
     case Normal: switch (C) {
          case Arrow:
               Move(C,D);
               return(OK);
          case Del:
               Delete(D);
               if (Empty()) State = Empty;
               return(OK);
          default:
               Insert(C,D);
               if (Full()) State = Full;
               return(OK);
               }
...
     }
```

Událostmi řízené systémy

Všechny moderní systémy jsou řízené událostmi

- Události jsou ve Windows reprezentovány zprávami

```
struct Message {
  int Msg; /* Message type */
  int Window; /* Message target */
  int Param1; /* Message type dependent */
  int Param2; /* Message type dependent */
};
```

- Zprávy jsou „zasílány“ voláním specializovaných funkcí exportovaných prvky rozhraní (typicky okny)

Vztah prvků rozhraní a objektů

- Prvky rozhraní jsou historicky (dosud) realizovány pomocí neobjektového API s „balíčky dat“
- Balíčky dat jsou reprezentovány „handly“, protože patří rozhraní a ne aplikaci
- Aplikace nemá přímý přístup do dat prvků rozhraní

Poznámka

- Výše uvedené skutečnosti jsou důsledkem faktu, že paměťový prostor aplikace je odlišný od paměťového prostoru modulu rozhraní (ať je to cokoliv) – data rozhraní nejsou přístupná aplikaci

Vztah prvků rozhraní a objektů

Vytvoření prvku rozhraní:

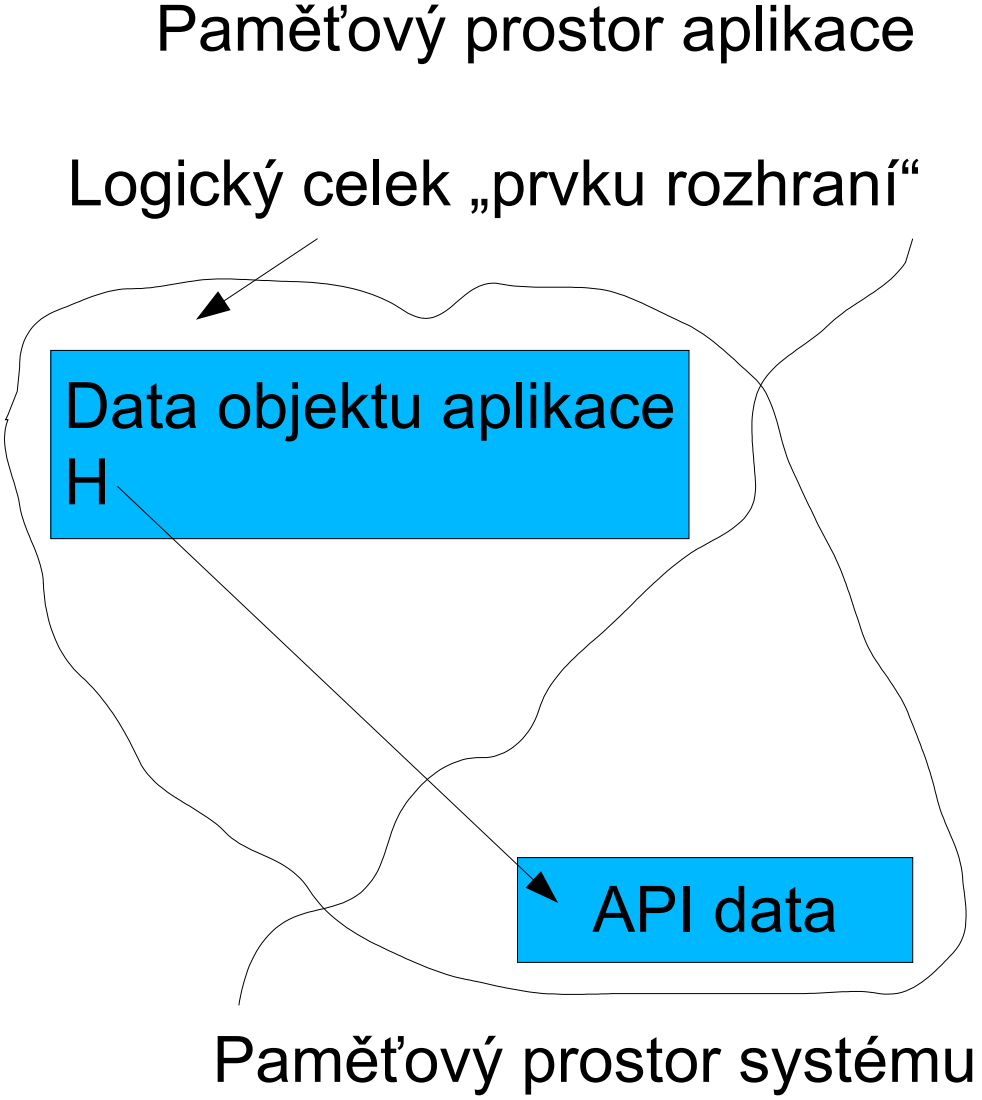
int CreateButton(...)

... H = CreateButton(...) ...

Užití prvku rozhraní:

... SetButtonText(H,“nic“); ...

Prvek rozhraní lze “zabalit” do objektu, ale nelze to přímo začleněním jeho dat do dat objektu



Shrnutí

- Neefektivní komunikace člověka s počítačem je jedním z nejzávažnějších problémů při využití počítčů
- Komunikace člověka s počítačem může probíhat pouze po velmi úzké množině vhodných komunikačních kanálů
- Pro komunikaci se vyrábí řada zařízení - i netradičních
- Komunikace se dnes realizuje pomocí systémů řízených událostmi
- K realizaci rozhraní se typicky užívají systémy s grafickým návrhem rozhraní

Příklady textů užité při přednášce

```
int HadleAll(Param)
{
  switch(Param->Msg)
  {
  case Keyboard:
    HadleKey(Param)...
    break;
  case Mouse:
    HandleMouse(Param)...
...
//  case ButtonPress:
//    PressButton(Message);
//    break;
//  ...
}
```

Ukázka kódu „WindowProc“ pro zpracování příchozích zpráv do okna

```
class Button...
...
Handle H;
...
SetText(char *)

-----------------------

Button * X = new Button(...)
...
X->SetText("adsfasd");

-----------------------

SetButtonText(X->H,"asdfasdf");
```

Ukázka kódu „obalujícího“ struktury Windows do objektu v C++ a volání API

Příklady textů užité při přednášce

```
class Button
{
...
virtual void PressButton();
...
}

class MyButton: public Button
{
virtual void PressButton();
...
}
```

Ideální stav pro „overloading“ virtuálních metod při objektovém programování

```
class Button: public Window
{
virtual int PressButton(Message)
virtual int MouseMove(Message)
...
}


class MyButton: public Button
{
virtual int PressButton(Message);
}
```

Takto by mohl kód vypadat pro objekty „obalující“ strukturu „okno“ Windows API

Příklady textů užité při přednášce

```
class Button: public Window
{
virtual int PressButton(Message)
virtual int MouseMove(Message)
...
}


class MyButton: public Button
{
virtual int PressButton(Message);
virtual int MyHandler(Msg);
// assigned NO message
}

//Language extension
...
virtual int MyHandler(Msg) [WM_MY];
...
```

Jak se ale „napojí“ uživatelská zpráva, o níž designer knihovny nevěděl?

```
int HandleAll(Msg)
{
  for (i = 0;i<Max;++i)
  {
    if (Table[i].Type==Msg.Type)
      Table[i].Method(Msg);
      return;
  }
}
```

Dnes užívaným řešením je tabulka obsahující dvojice „zpráva-metoda“