Začínáme s C/C++

Petr Šaloun

katedra informatiky FEI VŠB-TU Ostrava

26. září 2005

Základní pojmy

Algoritmus

- **jasný**,
- **rezultativní**,
- **konečný**.

Provádí **počítač**: CPU, sběrnice, paměť (hierarchie, vnitřní a vnější), I/O.
bit a **bajt**.
Operační systém
MS Windows 95, 98, ME, NT4.0, 2000, XP,
Unix – Linux, AIX, Solaris či Sun OS, HP-UX,
– **vstup a výstup**,
– **paměť**,
– **souborový systém**.
`muj.c` – **zdrojový text**
`muj.exe` **proveditelný program**
Soubory **textové** a **binární**.

Programovací jazyky

– **imperativní** – Ada, BASIC, C, C++, C#, FORTRAN, Java, Modula, Pascal, Smalltalk;
– **funkcionální** – Lisp, Haskell.
syntaxe – způsob zápisu,
sémantika – význam kódu.
Vývoj
60. léta FORTRAN, BASIC
domácí počítače (Sinclair, Commodore, Atari a Amiga),
osobních počítače (IBM PC, Macintosh),
Internet, hry, multimédia a zábava.
vyšší bezpečnost, typová kontrola, abstrakce, znovupoužitelnost: Simula, Smalltalk, Ada
systémové jazyky C, C++
von Neumannova architektura počítače,
sémantická mezera,
virtuální stroj.

Programovací jazyky C a C++

Bellovy laboratoře AT&T
Denis Ritchie, Brian Kernighan, Ken Thompson.
snadá a přenositelná implementaci Unixu
systémový jazyk, VŠ studenti → základ ISO normy – International Standards Organisation.

Vznik a vývoj C++

1986 Stroustrup: **The C++ Programming Language**, OOP, moduly.
90. léta
komponentní programování a Internet
Java a **C#**.

Principy objektově orientovaného programování

skutečnost $\longrightarrow$ programovací jazyk
pojmy **třída** a **objekt**
principy:
obalení – (encapsulation) spojení dat a metod (funkcí) do nové struktury – **třídy**;
dědičnost – (inheritance) nová třídy je dědicem tříd předků – specializace;
mnohotvárnost – (polymorphism) stejné pojmenování vlastností (metod) v hierarchii tříd, implementace se liší.

Obalení – abstrakce

Třída je popisem množiny objektů se společnými vlastnostmi.
Objekt je **instancí** třídy,
obsahuje (členská) **data** a **metody**.
Jedinečnost objektu.
Posílání zpráv = volání metod.
datová abstrakce – reprezentace dat je před uživatelem ukryta.

Dědičnost a hierarchie tříd

Dědičnost umožňuje rychlou tvorbu nových tříd znovupoužitím stávajícího kódu bez nutnosti jeho přepisování. Vzniká **hierarchie tříd**.
Pohyb dolů v této hierarchii obvykle znamená zvýšení **specializace** tříd, pohyb vzhůru naopak **zobecnění**.
předek (rodičovská třída, bázová třída, báze)
potomek (dceřinná třída, odvozená třída)
taxonomie či biologická analogie pokulhává – potomek může mít více než dva předky
Vícenásobná dědičnost přináší možné konflikty při opakovaném dědění ze společných předků.
doporučení: umírněné používání vícenásobné dědičnosti, případně její nahrazení jinými technikami (skládání).
Při tvorbě objektově orientovaného návrhu programu jsou kladeny velké nároky na správnou tvorbu hierarchie tříd. Čas strávený úvahami s tužkou v ruce ve fázi návrhu se bohatě zúročí.

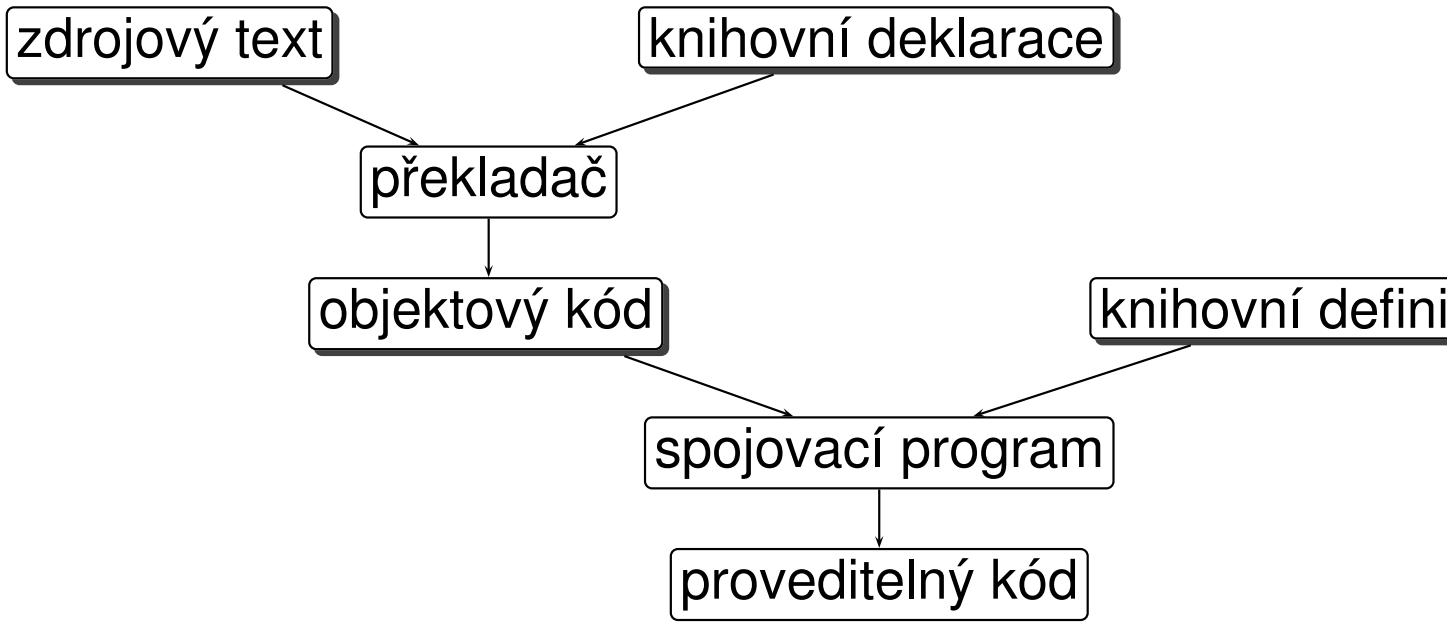
Mnohotvárnost – polymorfismus

Řecké slovo polymorfismus znamená mající mnoho tváří. Česky hovoříme o mnohotvárnosti. Použijeme-li v klasickém programu identifikátor funkce, víme vždy, když jej použijeme, jakou akci vyvolá. Spojení mezi identifikátorem a akcí se vytváří již v okamžiku překladu zdrojového textu. V C++ je mnohotvárnost těsně spjata s pojmem **virtuální metoda**. Taková metoda umožňuje použití více verzí téže metody v hierarchii tříd. Určení konkrétní metody, která bude použita, probíhá za běhu programu. brzda v automobilu

Překladač

Zdrojový text $\longrightarrow$ spustitelný tvar
Cílový procesor,
kód a data (kódový a datový segment),
spojovacím program,
knihovny funkcí a běhová podpora.
Chyba překladu (syntaktická),
chyba spojovacího programu,
chyba běhová (sémantická, jiná).

Překlad zdrojového textu v C++



Proč právě C++

- C/C++ je široce rozšířené a dostupné,
- dostupné jsou i ladicí nástroje a další podpůrné prostředky.
- Je vhodné pro rozsáhlé aplikace a má velmi blízko i ke strojové úrovni.
- Existují vyzrálé překladače
- Existuje norma jazyka.
- Moderní prvky OOP – obalení, dědičnost a mnohotvárnost.
- Moderní rysy, jakými jsou ošetření výjimek a prostor jmen.

PC, pracovní stanice, mnohaprocesorové superpočítače s možnostmi distribuovaného zpracování i PDA.
mýtus: programy v C++ jsou naprosto nečitelné
lék: správný programátorský styl zápisu zdrojového textu.

První programy

Ahoj, světe!

```
/**********************
 * hello.cpp
 * rychly zacatek
 **********************/

#include <iostream>
using namespace std; // prostor jmen

int main() {
 cout << "Ahoj svete!" << endl;
 return 0;
} // int main()
```

Srovnání stylů: klasický

```
/*****************************
 * zacatek-spatny.cpp
 * stary styl zacleneni hlavicek
 *****************************/

#include <iostream.h>

void main()
{
```

Srovnání stylů: moderní C++

```
/*****************************
 * zacatek-spravny.cpp
 * novy styl zacleneni hlavicek,
 * pouziva i prostory jmen
 *****************************/

#include <iostream>
using namespace std; // prostor jmen

int main() {
```

Jednoduchý vstup a výstup, simpleio 1-14

```
/*****************************
 * simpleio.cpp
 * jednoduchy vstup a vystup
 *****************************/

#include <iostream>
using namespace std;

int main() {
 cout << "zadej dve cela cisla:" << endl;
 int i, j;
 cin >> i >> j;
 cout << i << " + " << j << " = " << i+j << endl;
```

Jednoduchý vstup a výstup, simpleio 15-

```
    cout << "zadej racionalni cislo:" << endl;
    double x;
    cin >> x;
    double y = i * x;
    cout << i << " * " << x << " = " << y << endl;
    return 0;
} // int main()
```

```
zadej dve cela cisla:
2 3
2 + 3 = 5
zadej racionalni cislo:
6.78
2 * 6.78 = 13.56
```

Opakování části programu

cyklus
while
řídící podmínka cyklu
tělo cyklu
Vypočtěme a zobrazme hodnoty prvních deseti násobků zadaného celého čísla.

```
Zadej cele cislo: 7
   7  14  21  28  35  42  49  56  63  70
```

Opakování části programu, nasobky 1-14

```
/*********************************
 * soubor nasobky.cpp
 * vytiskne nasobky zadaneho celeho cisla
 *********************************/

#include <iostream>      // vstupy a vystupy
#include <iomanip> // formatovani vystupu

using namespace std;

int main() {
 const int od =  1; // dolni mez
 const int po = 10; // horni mez
 int cislo;
```

Opakování části programu, nasobky 14-

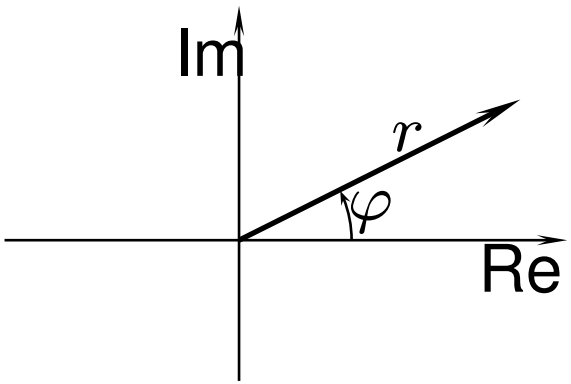
```
  cout << "Zadej cele cislo: ";
  cin >> cislo;

  int cinitel = od;
  while (cinitel <= po) {
      int soucin = cislo * cinitel;
      cout << setw(4) << soucin;
      cinitel = cinitel + 1;
  } // while (cinitel <= po)
  return 0;
} // int main()
```

Funkce a metody

Funkce v jazyce C++ mají jednoznačné jméno a mají určen počet a typ argumentů a typ návratové hodnoty.
návratový výraz: return ... ;
přetížení funkce – liší se její příjemce a tedy i definice, tj. způsob provedení.
Funkcím, které jsou součástí tříd, říkáme **metody**. S **daty** se pak komunikuje prostřednictvím metod, které jsou součástí rozhraní.
rozhraní třídy
public :

Geometrický význam třídy polar



Třída polar, tiskni 1-14

```
/**********************
 * soubor tiskni.cpp
 **********************/

#include <iostream>        // vstupy a vystupy
#include <string>          // retezcove definice a metody

using namespace std;

class polar {
        double r, fi;
public:
        polar(double a, double b)
                {r = a; fi = b;};          // konstruktor
```

Třída polar, tiskni 15-29

```
        double dej_r(void) {return r;};
        double dej_fi(void) {return fi;};
};

void tisk(int num) {
 cout << "cele cislo: " << num << endl;
} // void tisk(int num)

void tisk(double num) {
 cout << "racionalni cislo: " << num << endl;
} // void tisk(double num)

void tisk(string s) {
 cout << "retezec: " << s << endl;
} // void tisk(string s)
```

Třída polar, tiskni 31-50

```
void tisk(polar p) {
  cout << "r: " << p.dej_r()
       << " fi: " << p.dej_fi() << endl;
} // void tisk(polar p)

int main() {
 int i = 29;
 double x = 2003.07;
 string s = "kousek textu";
 polar y(65.43, 2.1); // tvorba objektu tridy polar

 tisk(i);
 tisk(x);
 tisk(s);
 tisk(y);
 return 0;
} // int main()
```

výstup tiskni

```
cele cislo: 29
racionalni cislo: 2003.07
retezec: kousek textu
r: 65.43 fi: 2.1
```