

Informatika 3

Kto je kto

Prednášajúci (a aj cvičiaci)

Viliam TAVAČ, Žilina
viliam.tavac@fri.uniza.sk
FRI A120

Cvičiaci

Miroslav Gábor
miroslav.gabor@fri.uniza.sk
FRI A117
František Kajánek
frantisek.kajanek@fri.uniza.sk
FRI A316
Andrej Šišila
andrej.sisila@fri.uniza.sk
FRI A316
Ján Šipikal
peter.novotny@fri.uniza.sk

Pomôcky pri štúdiu



Moodle

- Prezentácie
- Texty prednášok
- Riešené príklady z cvičení

Knihy

- **Thinking in C++**, 2nd Edition, Bruce Eckel,
<http://www.msedv.at/books/eckel/>
- **Programovací jazyk C++ 1., 2., 3. diel**, Miroslav Virius, Česká technika - ČVUT

Visual .net vývojové prostredie, help

Požiadavky na skúšku - Moodle

Získanie minimálne **25** bodov počas semestra.

- KT - kontrolné testy – riešenie zadaní na počítači
- AC - aktivita na cvičeniach

Skúška – test + riešenie úlohy na počítači



Programming Language

Ratings 2019/9

1	Java	16.661%
2	C	15.205%
3	Python	9.874%
4	C++	5.635%



C jazyk

- „vedľajší produkt“ vývoja operačného systému Unix 1969-1973
 - 1989 prijatý štandard ANSI-C, 1990 prijatý aj organizáciou ISO
 - C99 – prijatý ISO v 1999 a ANSI v marci 2000
 - C18 – jún 2018
- **Stručný**
 - **Kompaktný výkonný cieľový kód**
 - **Možnosť ovládať hardvér**
 - **Strojovo-nezávislý jazyk**
 - **Prenositeľnosť na rôzne hardvérové konfigurácie**



Dennis Ritchie

Filozofia programovania v C

Program = Dáta + Algoritmy

Dáta – informácia, spracovávaná programom

Algoritmy – metódy programu na spracovanie dát

Procedurálny jazyk



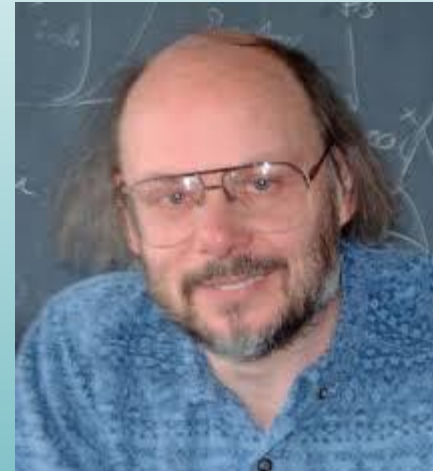
- Dôraz na algoritmickú stránku programovania
- Zhora nadol
- Funkcie (procedúry)

C++

- uľahčiť písanie dobrých programov



- 1983 – C++
- 1998 štandard ISO/IEC - ANSI C++ alebo C++98
- 2003 – aktuálna verzia C++03
- 2017 – december C++17 ISO štandard



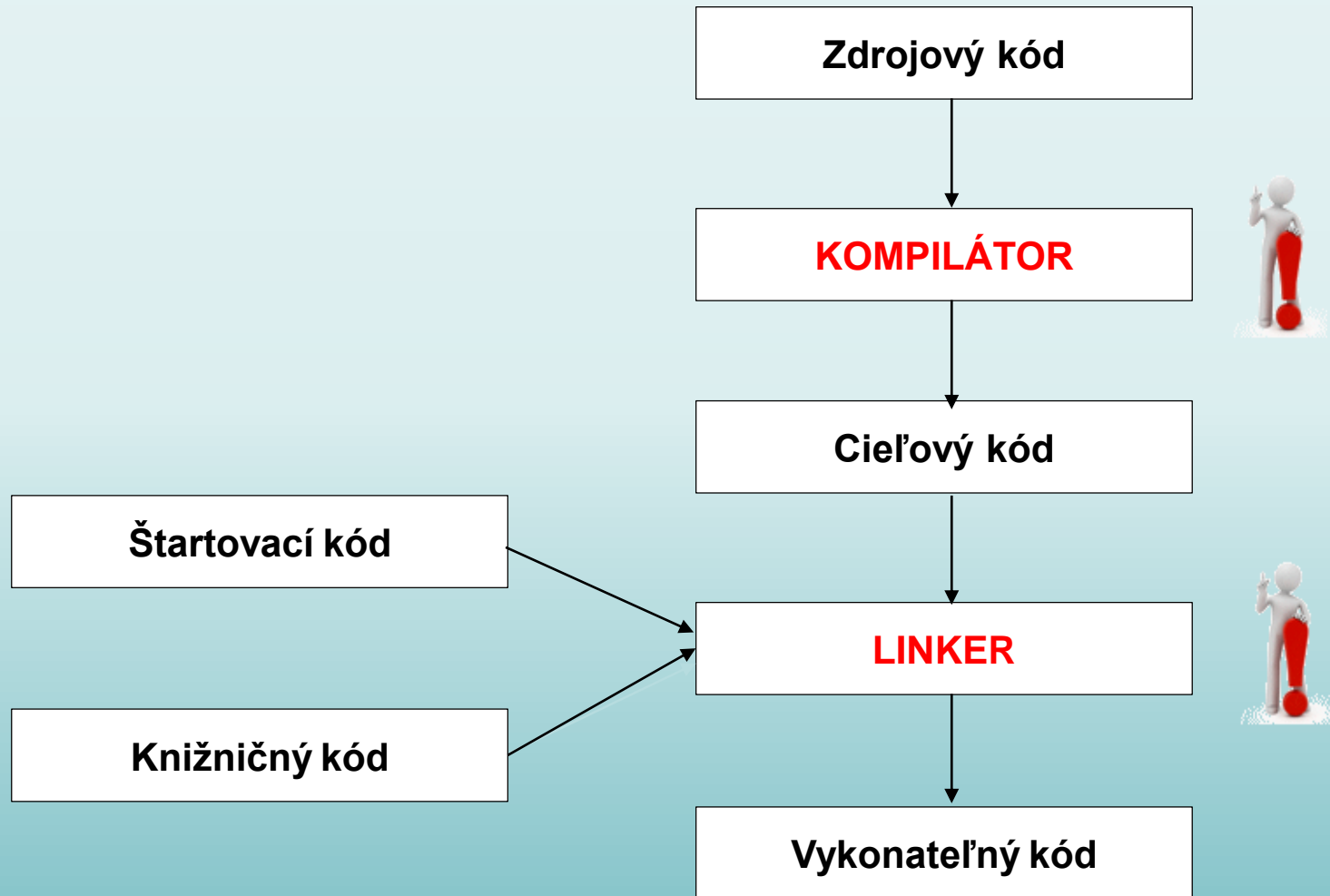
Bjarne Stroustrup

Filozofia programovania v C++

- Programovanie zdola nahor
- Generické programovanie - šablóny



Mechanizmus tvorby programu v C a C++



Proces prekladu



Súbor - zdrojový kód

- Ľubovoľný textový editor
- Sofistikované vývojové prostredie

- **Visual Studio 2019**

(<http://www.fri.uniza.sk/stranka/softver-a-internet>)



„...zaradenie do licenčného programu Microsoft DreamSpark (predtým MSDN AA), kde si študenti FRI bezplatne môžu sťahovať a inštalovať softvér Microsoft a to operačné systémy, vývojové prostredie a aplikácie - podľa návodu [msdn](#)“



Prípona zdrojového súboru

C++ implementácia	Prípona zdrojového súboru
Unix	C, cc, cxx, c
GNU C++	C, cc, cxx, cpp, c++, c
Microsoft Visual C++	cpp, cxx, cc, c



C a C++ program

// **prvy.c** - zobrazí oznam

```
#include <stdio.h>
```

```
int main()
```

```
{  
    printf("Ahoj C++.");  
    printf("\n");  
    printf("Zaciname!\n");  
    return 0;  
}
```

// **prvy.cpp** - zobrazí oznam

```
#include <iostream>
```

```
using namespace std;
```

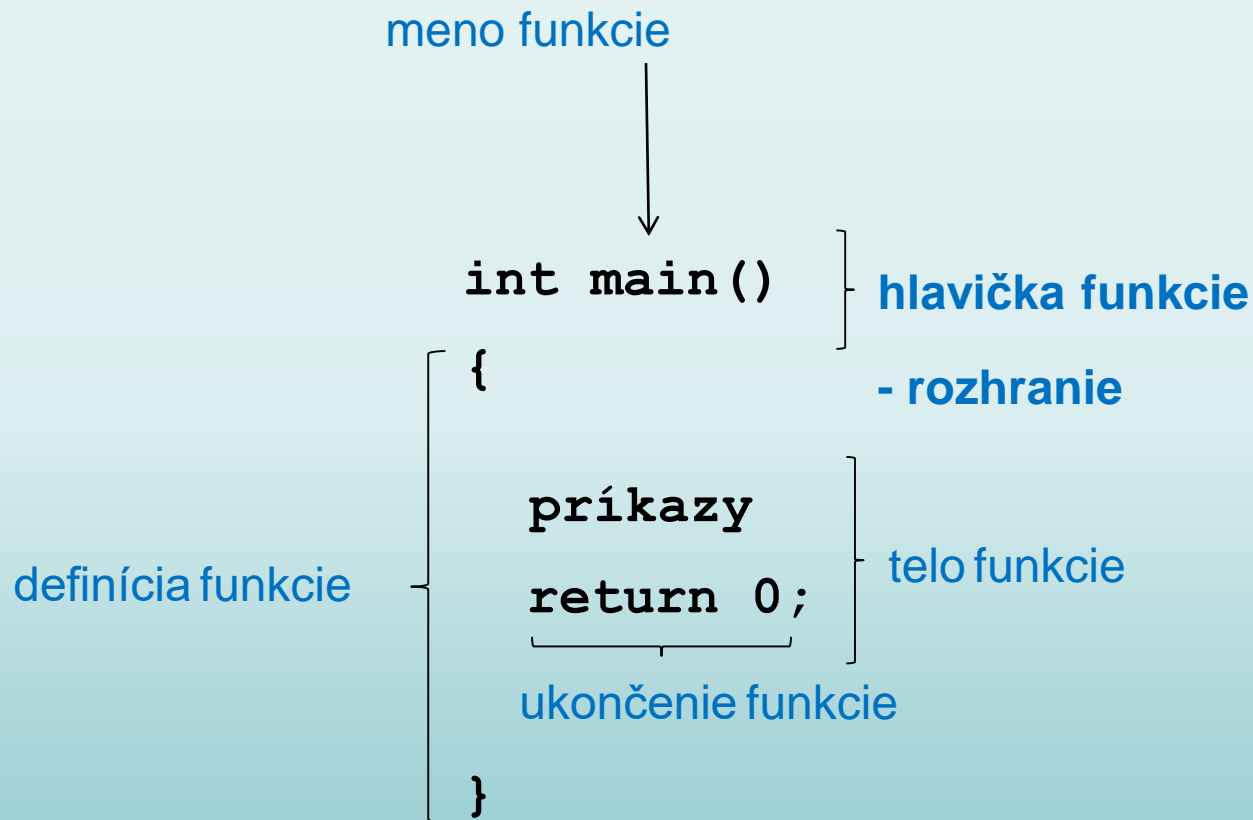
```
int main()
```

```
{  
    cout << "Ahoj C++.";  
    cout << endl;  
    cout << "Zaciname!" << endl;  
    return 0;  
}
```

```
Ahoj C++.  
Zaciname!  
Press any key to continue . . .
```



Funkcia main



```
int main(int argc, char *argv[])
```

Komentáre



```
#include <stdio.h>    /* a C komentár */
```

```
#include <iostream> // a C++ komentár (od C99 i v C)
```

Preprocesor



#include <stdio.h>

- program, ktorý spracováva zdrojový súbor pred vykonaním samotnej kompilácie
- direktívy preprocesora začínajú znakom #

Hlavičkové (včleňované) súbory

`stdio.h, iostream`

- včleňujú sa na začiatku do iných súborov
- príkaz preprocesora **#include**

`#include <standardny>`



alebo

`#include "vlastny.h"`



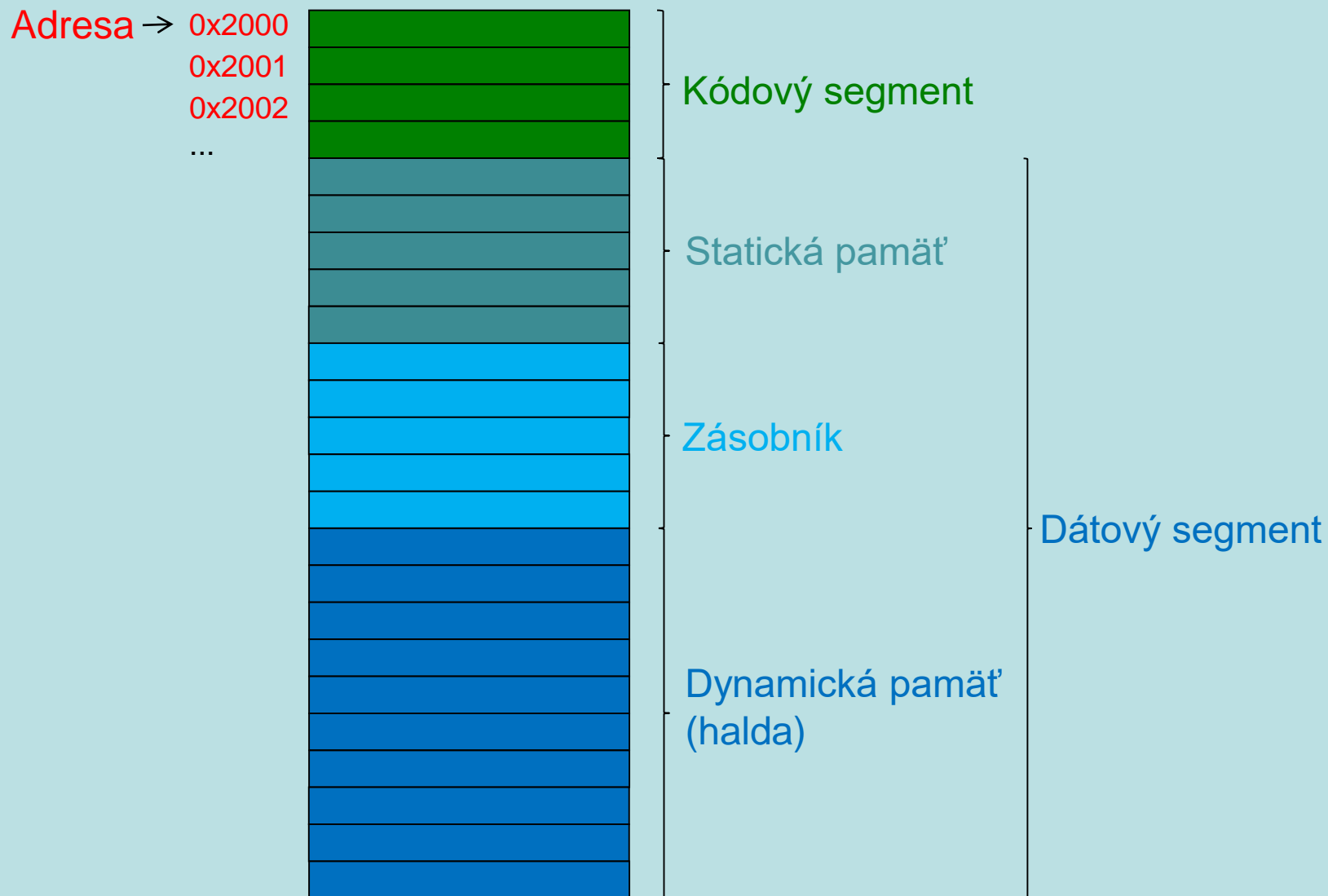
Hlavičkové súbory - prípony



`stdio.h`, `iostream`

Druh hlavičky	Konvencia	Príklad	Použiteľné v programoch
C++ starý štýl	Končí príponou <code>.h</code>	<code>iostream.h</code>	C++
C starý štýl	Končí príponou <code>.h</code>	<code>math.h</code>	C a C++
C++ nový štýl	Bez prípony	<code>iostream</code>	C++, používa namespace <code>std</code>
Konvertované C	C prefix, bez prípony	<code>cmath</code>	C++, môžu používať nie-C vlastnosti

Pamät'



Smerníky – čo je to?

- smerník je premenná, ktorá obsahuje adresu inej premennej
- pomocou smerníka môžeme ku premenným pristupovať tzv. nepriamo



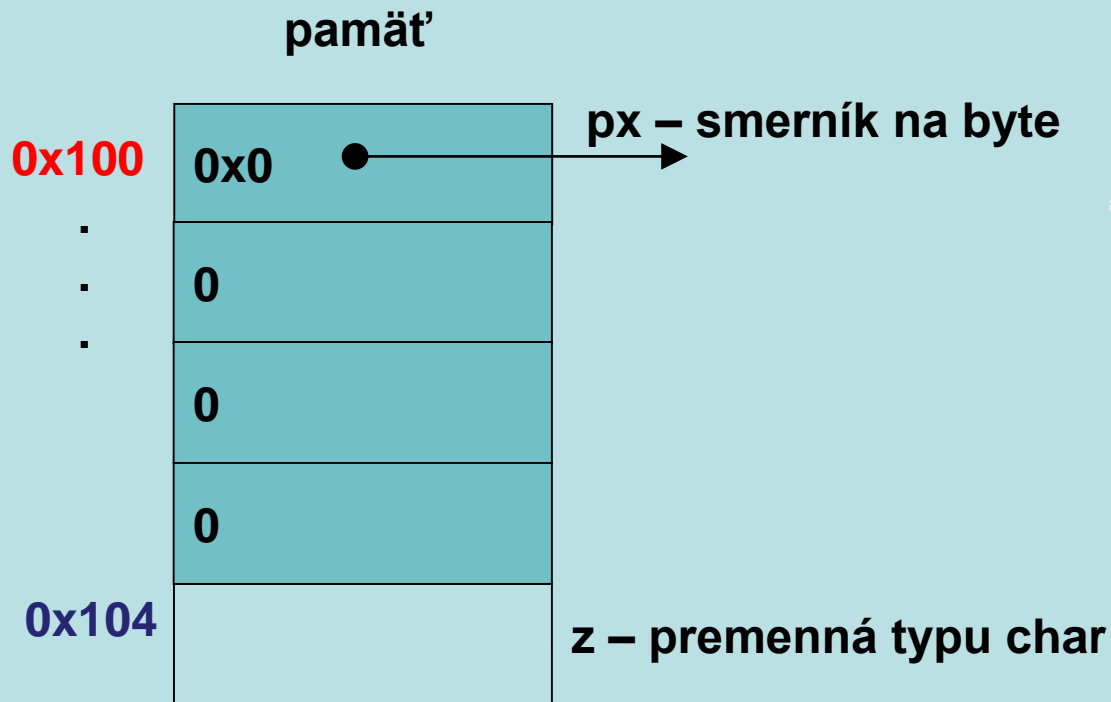
Smerníky - definícia

```
char *px;
```

sizeof(px) ... 4

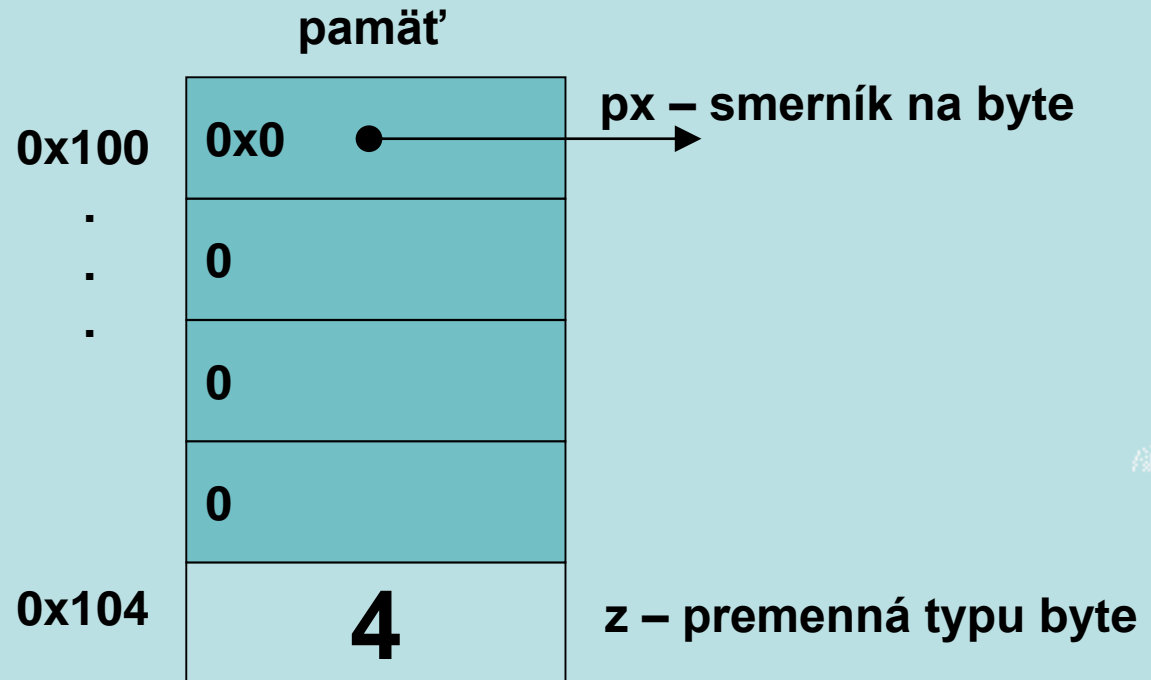
```
char z;
```

sizeof(z) ... 1

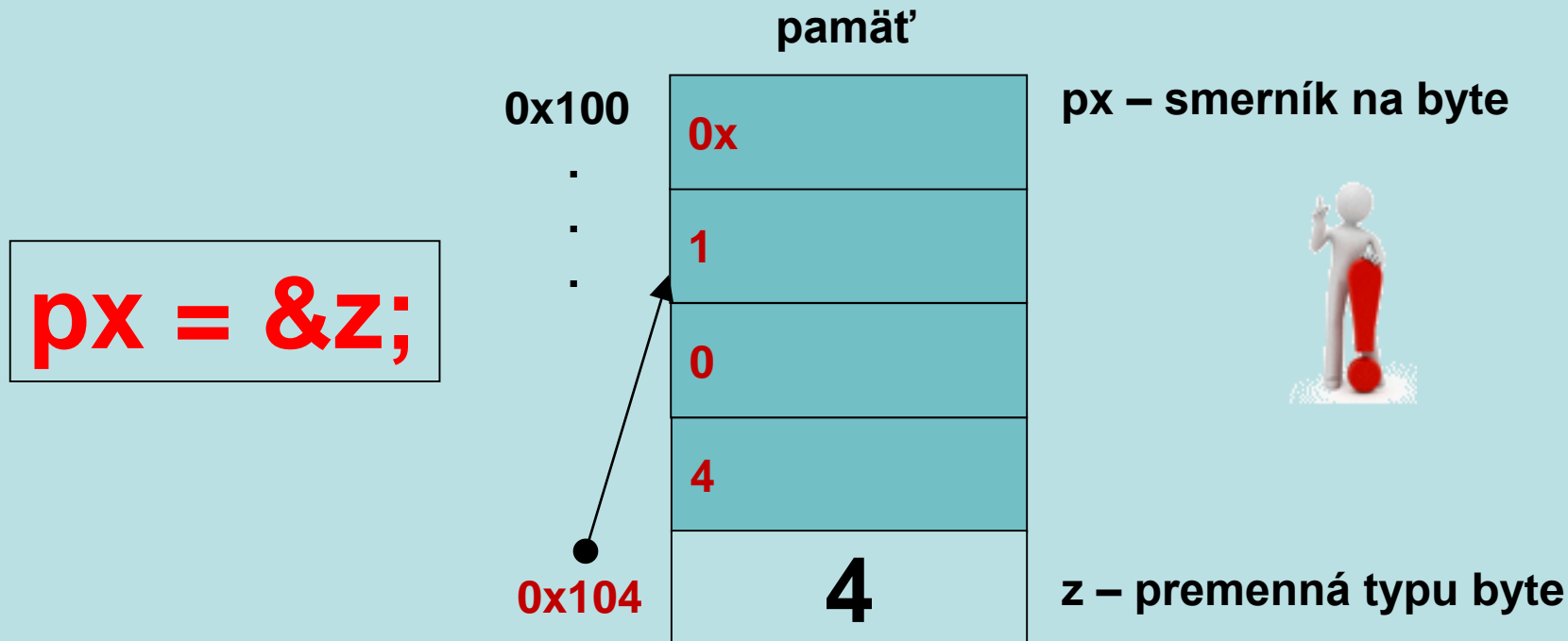


Smerníky – práca so smerníkom

z = 4;

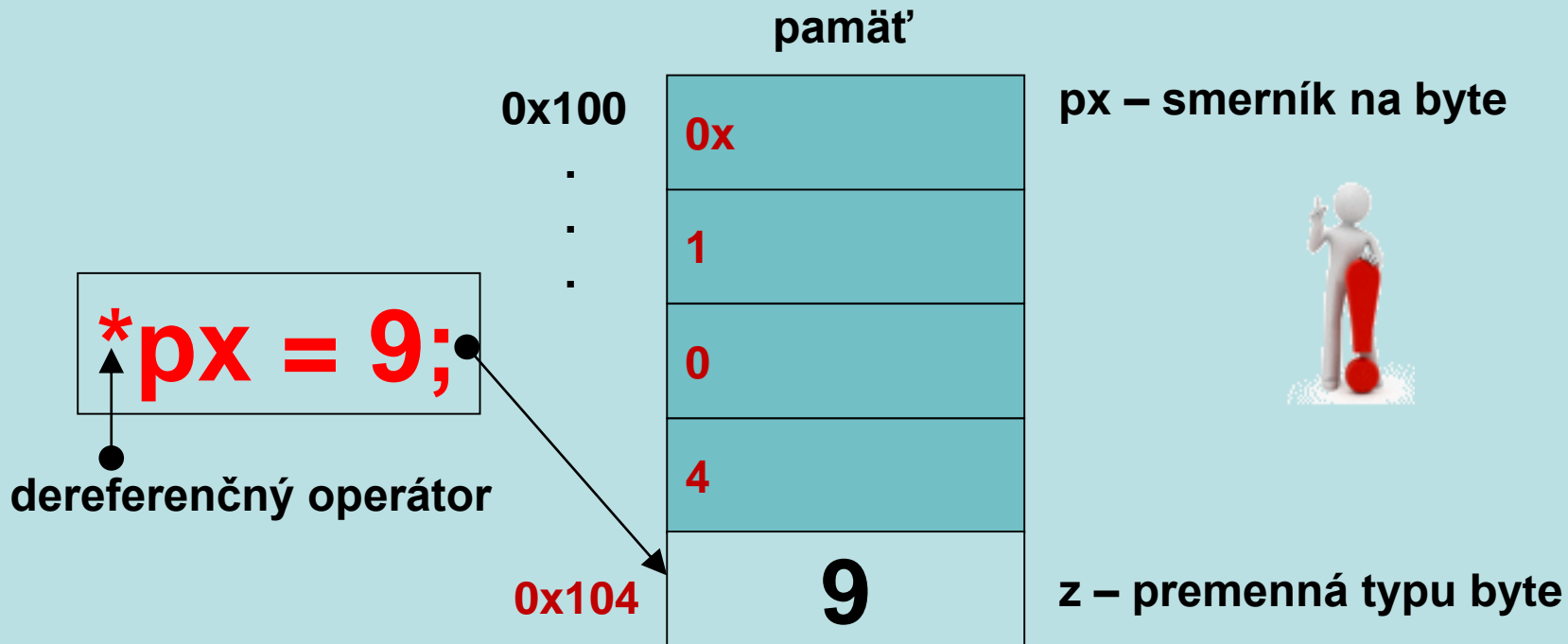


Smerníky – priradenie adresy



adresu premennej **z** získame výrazom **&z**

Smerníky – priradenie hodnoty



obsah premennej na ktorú ukazuje smerník **px**
získame/nastavíme výrazom ***px**

Smerníky

- Smerník môže ukazovať len na jeden konkrétny typ



Namespace - menopriestor

- Písanie rozsiahlych programov, ktoré kombinujú existujúci kód od rôznych tvorcov
- Štandardné knižnice v meno priestore **std**
- Direktíva **using**
 - `using namespace std;`
 - `std::cout`



```
// prvý.cpp - zobrazí oznam

#include <iostream>
// using namespace std;
int main()
{
    std::cout << "Ahoj C++.";
    std::cout << std::endl;
    return 0;
}
```