

Cvičenie 13

Obsah

- organizačné záležitosti
- príprava na skúšku
- zhrnutie

Organizačné záležitosti

- dostupnosť materiálov
- úlohy
 - odovzdanie
 - oprava
- zápočty

Skúška – 1. časť – záverečný príklad

- zadania
- autor
- dotazy
- prihlasovanie (po zápočte), schválenie
- rôzne bodovanie (náročnosť, bonusy)
- odovzdanie (týždeň pred skúškou)
- prípadná oprava, náhradný príklad, opravný termín

Skúška – 2. časť – odpovedník

- kde organizačné pokyny?
- skúškové termíny, prihlasovanie
- ako to prebieha
- časový limit, priebežné ukladanie
- počet správnych odpovedí
- bodovanie
- stratégia
- potrebný počet bodov (0 na E, 60 na A)
- šanca na úspech
- okruh otázok, štýl otázok

- skúšobná skúška -

- 20 otázok
- +2 body za správnu, -2 body za nesprávnu/žiadnu odpoveď
- -40 až +40 bodov
- nie úplná simulácia skúšky

Otázka 1

- nedefinované metódy
 - situácia: v *trieda.h* uvidíme v deklarácii triedy *trieda* metódu *trieda::metoda()*, ktorú chceme definovať v *trieda.cc...* a zabudneme na ňu (nedefinujeme ju, nevoláme ju)
 - otázka: čo sa stane?

Otázka 1

- nedefinované metódy
 - situácia: v *trieda.h* uvidíme v deklarácii triedy *trieda* metódu *trieda::metoda()*, ktorú chceme definovať v *trieda.cc...*a zabudneme na ňu (nedefinujeme ju, nevoláme ju)
 - otázka: čo sa stane?
 - odpoveď: nič
 - pokiaľ funkciu nevoláme, nevadí, že definícia chýba

Otázka 2

- postup prekladača pri analýze kódu
 - otázka: ako bude interpretovať $a+++b$; ?
 - možnosti:
 - výraz je chybný
 - $(a++) + b$;
 - $a + (++b)$;
 - $a + (+(+b))$;

Otázka 2

- postup prekladača pri analýze kódu
 - otázka: ako bude interpretovať $a+++b$; ?
 - možnosti:
 - výraz je chybný
 - $(a++) + b$;
 - $a + (++b)$;
 - $a + (+(+b))$;
 - správna odpoveď: $(a++) + b$;
 - prekladač postupuje od konca posledného nájdeného symbolu ďalej

Otázka 3

- komentáre C a C++
 - vid' úloha o zistení prekladača
- prázdny príkaz
 - `;`, `{ }`
 - otázka: ktorý z príkazov je správny?
 - `if (x>0) ; else cout << x;`
 - `if (x>0) { } else cout << x;`

Otázka 3

- komentáre C a C++
 - vid' úloha o zistení prekladača
- prázdny príkaz
 - `;, {}`
 - otázka: ktorý z príkazov je správny?
 - `if (x>0) ; else cout << x;`
 - `if (x>0) { } else cout << x;`
 - odpoveď: oba (aj keď neobratné:-))

Otázka 4

- majme příklad:

```
#include<iostream>
```

```
using namespace std;
```

```
struct A {A(){cout<<"A";} ~A(){cout<<"~A";}};
```

```
struct B : virtual A {B(){cout<<"B";} ~B(){cout<<"~B";}};
```

```
struct C : virtual A {C(){cout<<"C";} ~C(){cout<<"~C";}};
```

```
struct D : B,C {D(){cout<<"D";} ~D(){cout<<"~D";}};
```

```
int main(void){D d; return 0;}
```

- otázka: čo je na výstupe?

Otázka 4

- otázka: čo je na výstupe?
- odpoveď: $ABCD \sim D \sim C \sim B \sim A$

Otázka 5

```
struct struktura {double y;} funkcia(void) { /*...*/ }
```

- deklarácia návratového typu v deklarácii funkcie
- otázka: je to OK?

Otázka 5

```
struct struktura {double y;} funkcia(void) { /*...*/ }
```

- deklarácia návratového typu v deklarácii funcie
- otázka: je to OK?
- odpoveď: v C áno, v C++ nie

Otázka 6

```
int main() {int i=main(); return 0;}
```

- otázka: nekonečný program/chyba?

Otázka 6

```
int main() {int i=main(); return 0;}
```

- otázka: nekonečný program/chyba?
- odpověď: samozřejmě chyba

Otázka 7

- `break`;
 - ukončuje *for*, *while*, *do*, *switch*
 - otázka: čo sa stane, ak sa vyskytne mimo nich?

Otázka 7

- `break`;
 - ukončuje *for*, *while*, *do*, *switch*
 - otázka: čo sa stane, ak sa vyskytne mimo nich?
 - odpoveď: chyba pri preklade

Otázka 8

- `continue;`
 - skok na ďalšiu iteráciu cyklu (*for*, *while*, *do*)
 - otázka: čo sa stane, ak je použitý mimo cyklu?

Otázka 9

- `continue`;
 - skok na ďalšiu iteráciu cyklu (*for*, *while*, *do*)
 - otázka: čo sa stane, ak je použitý mimo cyklu?
 - odpoveď: chyba pri preklade
- otázka: čo sa stane, ak je použitý vo vnútri viacerých vnorených cyklov?

Otázka 9

- `continue`;
 - skok na ďalšiu iteráciu cyklu (*for*, *while*, *do*)
 - otázka: čo sa stane, ak je použitý mimo cyklu?
 - odpoveď: chyba pri preklade
- otázka: čo sa stane, ak je použitý vo vnútri viacerých vnorených cyklov?
- odpoveď: skok na ďalšiu iteráciu najvnútornejšieho cyklu

Otázka 10

$(x1 < \max ? x1 : x2) = 100;$

- otázka: prejde to? :)

Otázka 10

$(x1 < \max ? x1 : x2) = 100;$

- otázka: prejde to? :)
- odpoved': áno

Otázka 11

- ukazatele
 - otázka: je odčítanie ukazateľov platný výraz?

Otázka 12

- ukazatele
 - otázka: je odčítanie ukazateľov platný výraz?
 - odpoveď: áno, ak nejde o triedne ukazatele
 - otázka: čo sa stane pri dereferencii *void *v* ?

Otázka 13

- ukazatele
 - otázka: je odčítanie ukazateľov platný výraz?
 - odpoveď: áno, ak nejde o triedne ukazatele
 - otázka: čo sa stane pri dereferencii *void *v* ?
 - odpoveď: chyba
 - otázka: môžeme pričítať k ukazateľu číslo?

Otázka 13

- ukazatele
 - otázka: je odčítanie ukazateľov platný výraz?
 - odpoveď: áno, ak nejde o triedne ukazatele
- otázka: čo sa stane pri dereferencii *void *v* ?
- odpoveď: chyba
- otázka: môžeme pričítať k ukazateľu číslo?
- odpoveď: áno (posunutie o *n* miest v poli)

Otázka 14

- goto
 - silný nástroj, ale dávať na neho pozor!
 - predanie riadenia na príkaz označený návěstím
 - syntax:

```
// nejaký kod  
NAVESTIE: funkcia(a, b);  
// ďalší kod  
if (podmienka) goto NAVESTIE;
```

Otázka 14

- goto
 - otázka: môže byť návěstie deklarované v inej funkcii ako goto, ktoré sa na neho odvoláva?

Otázka 15

- goto
 - otázka: môže byť návěstie deklarované v inej funkcii ako goto, ktoré sa na neho odvoláva?
 - odpoveď: nie (ale môžeme to realizovať pomocou `<setjmp.h>`)
 - otázka: môže byť návěstie definované za goto?

Otázka 15

- goto
 - otázka: môže byť návěstie deklarované v inej funkcii ako goto, ktoré sa na neho odvoláva?
 - odpoveď: nie (ale môžeme to realizovať pomocou `<setjmp.h>`)
 - otázka: môže byť návěstie definované za goto?
 - odpoveď: áno (v tej istej funkcii)

Otázka 16

- otázka: môžeme použiť naraz `const` a `volatile`?

Otázka 16

- otázka: môžeme použiť naraz `const` a `volatile`?
- odpoveď: áno (konštantné, ale nestále objekty)

Otázka 17

```
#include<iostream>  
using namespace std;
```

```
int main(void) {  
    int i;  
    int i;  
    cin >> i;  
    return i;  
}
```

- otázka: čo sa stane?

Otázka 17

- otázka: čo sa stane?
- odpoveď: chyba pri preklade

Otázka 18

- výčet v C++

```
enum farba {zelena = 11, cervena = 22};
```

```
farba f;
```

– otázka: ktoré zápisy sú správne/nesprávne?

- `f = zelena;`
- `f = 11;`
- `f = (farba)11;`
- `f = farba(11);`

Otázka 18

- výčet v C++

```
enum farba {zelena = 11, cervena = 22};
```

```
farba f;
```

- otázka: ktoré zápisy sú správne/nesprávne?

- `f = zelena;`
- `f = 11;`
- `f = (farba)11;`
- `f = farba(11);`

- odpoveď: zápis `f = 11;` je nesprávny, ostatné sú správne

Otázka 19

- výčet v C

```
enum farba {zelena = 11, cervena = 22};
```

```
farba f;
```

– otázka: ktoré zápisy sú správne/nesprávne?

- `f = zelena;`
- `f = 11;`
- `f = (farba)11;`
- `f = farba(11);`

Otázka 19

- výčet v C

```
enum farba {zelena = 11, cervena = 22};
```

```
farba f;
```

- otázka: ktoré zápisy sú správne/nesprávne?

- `f = zelena;`
- `f = 11;`
- `f = (farba)11;`
- `f = farba(11);`

- odpoveď: v C je nesprávny aj posledný zápis (konštruktor)

Otázka 20

```
int a=1;
```

```
for(int b=0; b<=10; (++b)++){(++a)++;}
```

- otázka: akú hodnotu bude mať *a* na konci?
- možnosti:
 - 1
 - 11
 - 13
 - 23
 - iná hodnota

Otázka 20

```
int a=1;
```

```
for(int b=0; b<=10; (++b)++){(++a)++;}
```

- otázka: akú hodnotu bude mať *a* na konci?
- možnosti:
 - 1
 - 11
 - 13
 - 23
 - iná hodnota
- odpoveď: 13 :)

Ako ste dopadli?

Poznámky

- čo je vhodné pozrieť pred skúškou?
 - nezabudnúť skúsenosti nadobudnuté pri úlohách
 - materiály a demá z cvičení
 - céčkovské “skrčky”
 - <http://www.fi.muni.cz/usr/jkucera/pb161/sl10.htm>
 - rozdiely medzi C a C++
 - <http://www.fi.muni.cz/~xcupak/pb161.html>
 - prednášky k C++
 - <http://www.fi.muni.cz/usr/jkucera/pb161/>
 - prednášky k C (dost' veľa sa opakuje)
 - <http://www.fi.muni.cz/usr/jkucera/pb071/>

The end.