

# PB161

## 4. cvičení

Lukáš Ručka

14. října 2011

```
namespace geometrie {
    class vector {
        typedef float coord_t;

        vector(const coord_t x, const coord_t y);
        vector operator+(const vector & a) const ;

        coord_t x;
        coord_t y;
    };
}

namespace geometrie {
    vector(const vector::coord_t nx, const vector::coord_t ny)
        :x(nx),y(ny)
    { ; }
}

geometrie::vector geometrie::vector::operator+(const geometrie::
    vector & a) const {
    return vector(x+a.x, y+a.y);
}
```

```
namespace geometrie {
    class vector {
        typedef float coord_t;

        vector(const coord_t x, const coord_t y);
        vector operator+(const vector & a) const ;

        coord_t x;
        coord_t y;
    };
}

namespace geometrie {
    vector(const vector::coord_t nx, const vector::coord_t ny)
        :x(nx),y(ny)
    { ; }
}

geometrie::vector geometrie::vector::operator+(const geometrie::
    vector & a) const {
    return vector(x+a.x, y+a.y);
}
```

./chyba.cpp:10: error: '::main' must return 'int'

```
#ifndef MESSAGE_H
#define MESSAGE_H

#include <cstring>

struct message_data {
    size_t len;
    void * data;
};

class message {
public:
    message(){ ; }
    virtual ~message(){ ; }

    virtual message_data compose_message() = 0;
};

#endif // MESSAGE_H
```

```
int zvys(int cislo, int prirustek = 1){  
    return cislo+prirustek;  
}
```

```
int vynasob(int a, int b){  
    return a*b;  
}
```

```
short int vynasob(int a, int b){  
    return a*b;  
}
```

# Najdi chybu

PB161

Lukáš Ručka

Rekapitulace

Chyby

```
1 #include <iostream>
2
3 using namespace sld;
4
5 int main (int argc, long ** argv){
6
7     cout << argc << endl;
8     return 0;
9
10 }
```

# Najdi chybu

PB161

Lukáš Ručka

Rekapitulace

Chyby

```
1 #include <iostream>
2
3 using namespace sld;
4
5 int main (int argc, long ** argv){
6
7     cout << argc << endl;
8     return 0;
9
10 }
```

Chyba je na řádku:

A) 3   B) 5   C) 7   D) 3, 5   E) 3, 7   F) nikde

```
1 #include <iostream>
2
3 using namespace sld;
4
5 int main (int argc, long ** argv){
6
7     cout << argc << endl;
8     return 0;
9
10 }
```

Chyba je na řádku:

A) 3   B) 5   C) 7   **D) 3, 5**   E) 3, 7   F) nikde



# Najdi chybu

```
1  #include <iostream>
2
3  class A {
4  public:
5      A (float l):c(l){ ; } ;
6      float c;
7  };
8
9  using namespace std;
10
11 void main (int argc, const char ** argv){
12
13     A A(4.5f);
14
15     cout << A.c << endl;
16
17 }
```

# Najdi chybu

```
1  #include <iostream>
2
3  class A {
4  public:
5      A (float l):c(l){ ; } ;
6      float c;
7  };
8
9  using namespace std;
10
11 void main (int argc, const char ** argv){
12
13     A A(4.5f);
14
15     cout << A.c << endl;
16
17 }
```

Chyba je na řádku:

- A) 5    B) 11    C) 13    D) 11, 13    E) 13, 5    F) nikde

# Najdi chybu

```
1 #include <iostream>
2
3 class A {
4 public:
5     A (float l):c(l){ ; } ;
6     float c;
7 };
8
9 using namespace std;
10
11 void main (int argc, const char ** argv){
12
13     A A(4.5f);
14
15     cout << A.c << endl;
16
17 }
```

Chyba je na řádku:

- A) 5    B) 11    C) 13    **D) 11, 13**    E) 13, 5    F) nikde

# Najdi chybu

```
1  #include <iostream>
2
3  class A;
4  class B;
5
6  struct C {
7      B * get_value(const A & a);
8  }
9
10 using namespace std;
11
12 int main (int argc, const char ** argv){
13     A a;
14
15     C cc;
16     cout << cc.get_value(a) << endl;
17
18     return 0;
19 }
```

# Najdi chybu

```
1 #include <iostream>
2
3 class A;
4 class B;
5
6 struct C {
7     B * get_value(const A & a);
8 }
9
10 using namespace std;
11
12 int main (int argc, const char ** argv){
13     A a;
14
15     C cc;
16     cout << cc.get_value(a) << endl;
17
18     return 0;
19 }
```

Je zde opravdu chyba?

# Najdi chybu

```
1 #include <iostream>
2
3 class A;
4 class B;
5
6 struct C {
7     B * get_value(const A & a);
8 }
9
10 using namespace std;
11
12 int main (int argc, const char ** argv){
13     A a;
14
15     C cc;
16     cout << cc.get_value(a) << endl;
17
18     return 0;
19 }
```

Je zde opravdu chyba? **Je!**