

# Písomná skúška z PT: 7. februára 2022

Táto písomná skúška trvá 90 minút. Počas skúšky je povolené používať knihy, poznámky, učebné texty, referencie jazyka C/C++, kompilátor jazyka C/C++. Kódy predpokladajú kompilátor s podporou C++ 17. Kódy boli testované s `g++ -std=c++17` (g++ verzia 9.3.0).

Túto písomnú skúšku je potrebné vypracovať samostatne, teda bez pomoci niekoho iného a bez komunikácie s niekým iným. Odhalené podvádzanie pri skúške, napr. kopírovanie odpovedí a riešení (aj ich častí), snaha odovzdať cudzie odpovede a riešenia a pod., môže byť penalizované vylúčením študenta z predmetu, nepridelením a/alebo zrušením bodov, a to aj bodov z cvičení, prípadne aj disciplinárnym konaním v zmysle Študijného poriadku, ktoré môže viesť k vylúčeniu zo štúdia.

## 1. Prečo kód neskompiluje? (6 b.)

```
1 #include <memory>
2
3 class Token {};
4
5 int main() {
6     std::unique_ptr<Token> ptr = std::make_unique<Token>();
7
8     delete ptr;
9
10    return 0;
11 }
```

---

## 2. a) Prečo kód neskompiluje? b) Čo je potrebné zmeniť v definícii triedy Token, aby kód skompiloval (bez zakomentovania akejkoľvek časti kódu)? (6 b.)

```
1 #include <vector>
2 #include <memory>
3
4 class Token {
5 private:
6     std::unique_ptr<std::vector<int> > ptr = std::make_unique<std::vector<int> >();
7 public:
8     Token() = default;
9     Token(Token&& t0) noexcept { ptr = t0.ptr; }
10 };
11
12 int main() {
13     Token t = Token();
14
15     return 0;
16 }
```

---

## 3. Prečo kód vypíše hodnotu 2? Koľko inštancií `std::vector<int>` vznikne? (8 b.)

```
1 #include <vector>
2 #include <memory>
3 #include <iostream>
4
5 class Token {
6 private:
7     std::shared_ptr<std::vector<int> > ptr = std::make_shared<std::vector<int> >();
8 public:
9     Token() = default;
10    Token(Token&& t0) noexcept { ptr = t0.ptr; }
11    auto getPtr() { return ptr; }
12 };
13
14 int main() {
15     Token t = Token();
16     std::cout << t.getPtr().use_count();
17
18     return 0;
19 }
```

---

4. Prečo nastane nedefinované správanie (v testovacom prípade boli vypísané hodnoty 11 a 21973)? (8 b.)

```
1 #include <iostream>
2
3 auto add_one(int init) {
4     auto f = [&init]() -> int {
5         return ++init;
6     };
7     return f;
8 }
9
10 int main() {
11     auto sum = add_one(10);
12     std::cout << sum(); //11
13     std::cout << sum(); //21973
14
15     return 0;
16 }
```

---

5. Čo implementuje tento kód (čo sa vypíše na riadku 14)? Čo implementuje použitý lambda výraz? (8 b.)

```
1 #include <vector>
2 #include <iostream>
3 #include <numeric>
4
5 int main() {
6     std::vector<int> v{1, 6, 7, 50, 99, -5};
7
8     double a = std::accumulate(v.begin(),
9                               v.end(),
10                              0.0,
11                              [&v](const double a, const double b) { return a + b/v.size(); }
12                              );
13
14     std::cout << a;
15
16     return 0;
17 }
```

---

6. a) Prečo tento kód neskompiluje? b) Čo je v kóde potrebné doplniť, aby kód skompiloval? (8 b.)

```
1 #include <map>
2 #include <iostream>
3
4 class Token {
5 private:
6     int a{};
7 public:
8     Token(int a0) : a(a0) {}
9 };
10
11 int main() {
12     std::map<int, Token> m;
13
14     for(int i=0; i<10; ++i) { m.emplace(i, Token{i}); }
15
16     for(auto x : m) { std::cout << x.first << "_" << x.second << std::endl; }
17
18     return 0;
19 }
```

---

7. a) Prečo kód neskompiluje? b) Ako je kód potrebné doplniť, aby skompiloval? (8 b.)

```
1 #include <vector>
2 #include <iostream>
3
4 template <typename T>
5 class TokenB {
6 private:
7     int a{1};
8     std::vector<T> v;
9 public:
10    TokenB operator%(const TokenB& t0) {
11        TokenB tmp;
12        tmp.a = a + t0.a;
13        return tmp;
14    }
15 };
16
17 template <typename T>
18 class TokenD : public TokenB<T> {};
19
20 int main() {
21     TokenB<int> t0;
22     TokenB<int> t1;
23
24     TokenD<int> t = t0 % t1;
25
26     return 0;
27 }
```

---

8. Prečo potrebujeme lambda výrazy f0 a f1? Aký majú význam? Aký význam má premenná m? Aký význam má decltype(f0) a decltype(f1)? (8 b.)

```
1 #include <unordered_set>
2 #include <iostream>
3
4 int main() {
5     const int m = 3;
6     auto f0 = [=](const int a) -> int {
7         return a % m;
8     };
9
10    auto f1 = [](const int a, const int b) -> bool {
11        return a == b;
12    };
13
14    std::unordered_set<int, decltype(f0), decltype(f1)> s(m, f0, f1);
15    s.insert({1, 6, 7, 50, 99, -5});
16
17    for(int x : s) { std::cout << x << std::endl; }
18
19    return 0;
20 }
```

---