2. Zárthelyi dolgozat

1. Definiáljunk egy `DestructCount` osztályt, amely egy `private static` változóban tárolja, hogy hány példányának futott le a destruktora a program futása során. Inicializáljuk a változót, és írjuk meg az osztály destruktorát is. Írjunk még egy `static Count()` tagfüggvényt is, amellyel egy felhasználó lekérdezheti, hogy hányszor futott le a destruktor. Más tagfüggvényt/tagváltozót nem kell deklarálni. (*10 pont*)

2. Írjunk egy függvényt, amely paraméterül kap egy `string` változót, és visszaadja a `007` karaktersorozat második előfordulásának indexét ebben a `string`-ben, ha pedig a karaktersorozat kettőnél kevesebbszer fordul elő, akkor visszaadja a `string::npos` értéket. (*10 pont*)

3. Definiáljunk egy `Container` nevű template osztályt, amelynek egyetlen tagváltozója egy `data` nevű `vector`, amely a template paraméterrel megegyező típusú elemeket tárol. Definiáljunk egyetlen tagfüggvényt, amely egy `size_t` típusú indexet kap paraméterként, és visszaad egy konstans referenciát, amely a `data` tagváltozó megfelelő elemére mutat. (Több tagfüggvényt deklarálni sem kell, tehát az adatvektorba való írást nem kell megvalósítani, csak a lekérdezést.) A függvény dobjon `std::invalid_argument` kivételt, ha nem megfelelő a paraméterként megadott index. (*12 pont*)

4. Mit ír ki a képernyőre következő kód és miért? (*10 pont*)

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

class base {
public:
    void fun_1() { cout << "base-1\n"; }
    virtual void fun_2() = 0;
    virtual void fun_3() { cout << "base-3\n"; }
};

class derived : public base {
public:
    void fun_1() { cout << "derived-1\n"; }
    void fun_2() { cout << "derived-2\n"; }
};

int main() {
    derived obj1;
    base* p = &obj1;

    p->fun_1();
    p->fun_2();
    p->fun_3();

    return 0;
}
```

5. Tegyük fel, hogy a `fun` függvény egy `int` típusú változót kap paraméterként, és azt is ad vissza. A függvényről azt is tudjuk, hogy hibás futás vagy rossz paraméter esetén egy `int` típusú kivételt dob (a függvény tényleges működése pedig a feladat szempontjából lényegtelen). Írjunk egy `try-catch`-blokkot, ami futtatja a függvényt valamilyen tetszőleges paraméterrel, és elkapja a kivételt, ha ilyet dob a függvény, valamint ebben az esetben kiírja a hibakódot, azaz a dobott kivétel értékét. (*8 pont*)