5. labor kis ZH, 2023. május 25., csütörtök 10-12

NÉV (olvasható): NEPTUN kód:

max. 10p, megoldási idő: 15 perc

1. Adott az alábbi Window osztály, amely egy téglalap alakú területet reprezentál a képernyőn. Származtass belőle szerkesztőablakot reprezentáló Editor osztályt, amely pluszként a szerkesztett szöveget tárolja! Készíts konstruktort, amely a szerkesztőablak minden adatát (a Window-hoz tartozókat is) inicializálja! Definiáld felül a print függvényt úgy, hogy az a szerkesztőablak minden adatát írja ki! Írj kódrészletet, amelyben létrehozol egy Editort, és kiírod az adatait! A Window osztályt nem módosíthatod! (4p)

```
class Window {
    int x, y, w, h;
public:
    Window(int x, int y, int width, int height)
        : x{ x }, y{ y }, w{ width }, h{ height } {}
    void print()const {
        std::cout << "x=" << x << ", y=" << y << ", w=" << w << ", h=" << h;
    }
};

class Editor :public Window { // 1p
    std::string text;
public:
    Editor(int x, int y, int width, int height, std::string text)
        : Window(x, y, width, height), text{ text } {} // 1p
    void print()const {
        Window::print(); // 1p
        std::cout << ", text: " << text;
    }
};

    Editor  g(1, 2, 3, 4, "OK"); // 1p
    g.print();
```

2. Készíts generikus függvényt, amely egy tömböt kap paraméterként, és visszaadja a tömb elemeinek összegét! Készíts specializációt C sztringekre mutató pointerek tömbje esetére, amely a leghosszabb sztring címét adja vissza! Írj kódrészletet, amelyben kipróbálod a függvényt double tömbre és C sztring-pointer tömbre is! (A kipróbálásnál felhasználhatod a következő sztringpointer tömböt.) (6p)

```
const char* sptomb[4] = { "Dénes", "Aladár", "Cecil", "Béla" };

template<typename T> // 1p
T osszead(const T t[], int n) { // 1p
    T sum = T();
    for (int i = 0; i < n; i++) // alg: 1p
        sum += t[i];
    return sum;
}

template<>
const char* osszead(const char* const t[], int n) { // specializáció 1p
    int leghosszabbIndex = 0;
    int leghosszabbHossz = strlen(t[0]);
    for (int i = 1; i < n; i++)
        if (strlen(t[i]) > leghosszabbHossz) {// alg 1p
            leghosszabbHossz = strlen(t[i]);
            leghosszabbIndex = i;
        }
    return t[leghosszabbIndex];
}
```

```
double dt[6] = { 10, -8, 0, 99, -11, 3 };
const char* sptomb[4] = { "Dénes", "Aladár", "Cecil", "Béla" };
std::cout << osszead(dt, 6) << std::endl; // ha mindkét példa jó: 1p
std::cout << osszead(sptomb, 4) << std::endl;
```