

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1 Complex osztály előadásról
 - ◆ 1.2 Téglalap
 - ◆ 1.3 Vissza a komplexre
 - ◆ 1.4 String osztály
 - ◆ 1.5 Láncolt lista

Feladatok

Minden feladathoz nyiss új projektet IDE-ben vagy írd új file-ba ha parancssorból dolgozol!

Mostantól .cpp kiterjesztésű file-okban dolgozzunk!

Complex osztály előadásról

Futassuk le az előadáson látott Complex osztályt:

```
#include<iostream>
#include<cmath>

using namespace std;

class Complex {
private:
    float re;
    float im;
public:
    Complex();
    Complex(const Complex& other);
    Complex(float r);
    Complex(float r, float i);

    Complex add(Complex other);
    Complex times(Complex other);
    float abs();

    void print();

    ~Complex();
};

Complex::Complex() {
    re = 0;
    im = 0;
}

Complex::Complex(const Complex& other) {
    re = other.re;
    im = other.im;
}
```

```
Complex::Complex(float r) {
    re = r;
    im = 0;
}

Complex::Complex(float r, float i) {
    re = r;
    im = i;
}

Complex Complex::add(Complex other) {
    return Complex(this->re + other.re, this->im + other.im);
}

Complex Complex::times(Complex other) {
    float real = this->re * other.re - this->im * other.im;
    float imag = this->re * other.im + this->im * other.re;
    return Complex(real, imag);
}

float Complex::abs() {
    return sqrt(this->re * this->re + this->im * this->im);
}

void Complex::print() {
    cout << re << " + " << im << "i" << endl;
}

Complex::~Complex() {
}

int main(void) {
    Complex a;
    Complex b = Complex(1,2);
    Complex c = a.times(b);

    a.print();
    b.print();
    c.print();

    (b.add(c)).print();

    cout << b.abs() << endl;

    return 0;
}
```

Téglalap

Készítsünk egy téglalap osztályt!

- Létrehozáskor kelljen megadni az oldalak hosszát.
- Tudjuk lekérdezni a területét és kerületét.
- Legyen egy **scale** metódusa, amivel nagyítani tudjuk (adott számmal szorzódik mind2 oldal).
- Lehesen másolni így (copy konstruktor):

```
Tegla a(2, 5);
Tegla b(a);
```

Vissza a komplexre

Nézzünk vissza a Complex osztályra. Etesük meg a kódot a pythontutor-al és kövessük végig, hogy mikor melyik metódus, konstruktor hívódik meg.

String osztály

Írjunk String osztályt, amivel kényelmesebben tudunk stringeket kezelni.

- Default konstruktor üres string-et hozzon létre
- Lehesen C string-ből konstruálni, ekkor a konstruktor fejének így kell kinéznie:

```
String(const char* s);
```

- Lehesen lekérni a hosszát
- Legyen print metódus, ami kiírja a string-et
- Lehesen két String-et konkatenálni

Láncolt lista

Valósítsuk meg a korábban használt láncolt listát mostmár osztályokkal. Tároljon a lista float-okat. A következők működjenek:

- default konstruktor: üres lista
- copy konstruktor
- konstruktor ami tömbből építi fel a listát
- destruktork
- append: új elem berakása
- insert: új elem adott helyre berakása
- size: tárolt elemek számát adja vissza
- in: az adott elem benne van-e a listában
- index: adott indexű elemet adja vissza
- remove: adott indexű elemet törli a listából
- where: visszaadja az első indexet ahol előfordul az adott érték, -1-et, ha nincs a listában