

Programação Orientado a Objeto

o mínimo para compreender a STL

Laboratório de Programação Competitiva I

Rene Pegoraro

Pedro Henrique Paiola

Wilson M Yonezawa

Classes

```
#include <bits/stdc++.h>
using namespace std;

struct Complexo {
    float r, i;

    void atrib(float _r, float _i) {
        r = _r; i = _i;
    }
};

int main() {
    Complexo a;

    a.atrib(1.2, 3.4);

    cout << a.r << ", " << a.i << endl;
    return 0;
}
```

```
#include <bits/stdc++.h>
using namespace std;
```

```
struct Complexo {
    float r, i;

    Complexo(float _r,
              float _i) {
        r = _r;
        i = _i;
    }
};

int main() {
    Complexo a(1.2, 3.4);
    Complexo *b = new Complexo(5.6, 7.8);

    cout << a.r << ", " << a.i << endl;
    cout << b->r << ", " << b->i << endl;

    delete b;

    return 0;
}
```

Operadores

```
#include <bits/stdc++.h>
using namespace std;

struct Complexo {
    float r, i;

    Complexo(float _r, float _i) {
        r = _r; i = _i;
    }

    void imprime() const {
        cout << r << ", " << i << endl;
    }
};

int main() {
    Complexo a(1.2, 3.4);
    a.imprime();
    return 0;
}
```

```
#include <bits/stdc++.h>
using namespace std;

struct Complexo {
    float r, i;

    Complexo(float _r, float _i) {
        r = _r; i = _i;
    }

    Complexo & operator+=(const Complexo &c) {
        r += c.r; i += c.i;
        return *this;
    }
};

int main() {
    Complexo a(2.1, 4.3);
    Complexo b(1.2, 3.4);

    a += b;

    cout << a.r << ", " << a.i << endl;
    return 0;
}
```