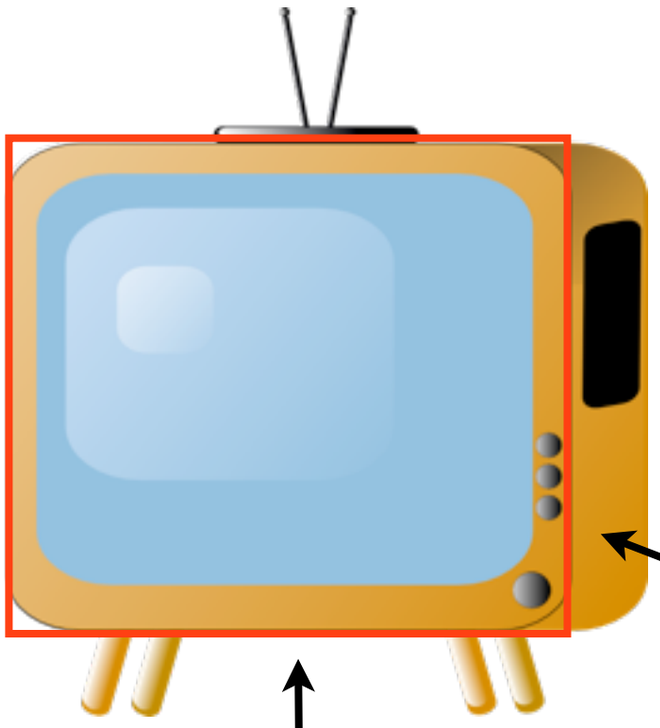
Programowanie

Dariusz Wardecki, wyk X

Klasy

Telewizor



Elektronika
w środku

Obudowa

Konkretne egemplarze:

Interfejs

Klasy

```
class Telewizor{
int opornik1;
int opornik2;
...
double kondensator1;
double kondensator2;
...
public:
void standby(){
...
}
void volume(int n){
...
}
};
```

```
int main()
{
  Telewizor mojTele;
  Telewizor* twojPilot;
  twojPilot = &mojTele;
  ...
  Telewizor* tv = new Telewizor;
  ...

  mojTele.volume(5);
  twojPilot->standby();
  tv->opornik1 = 100; //Blad !!!
}
```

Klasy vs struktury

```
struct Telewizor{
int opornik1;
int opornik2;
...
double kondensator1;
double kondensator2;
...
};
```

≡

```
class Telewizor{
 public:
  int opornik1;
  int opornik2;
  ...
  double kondensator1;
  double kondensator2;
  ...
};
```

Klasy i metody

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

class Prostokat {
    int x, y;
  public:
    void boki (int,int);
    int powierzchnia () {
    return (x*y);
    }
};
void Prostokat::boki (int a, int b) {
  x = a;
  y = b;
}
int main () {
  Prostokat rect;
  rect.boki (3,4);
  cout << "powierzchnia: " << rect.powierzchnia();
  return 0;
}
```

prototyp

metoda w ciele klasy

metoda poza klasą

Metody a "zwykłe" funkcje

```
class Klasa{
 public:
  int x, y;
  void metoda1();
  int metoda2(){
  return (x*y);
  }
};
void Klasa::metoda1(){
  ...
}
int funkcja(){
  return ...;
}

main(){
  ...
  Klasa obiekt;
  obiekt.x = 2;
  obiekt.y = 6;
  obiekt.metoda1();
  int y = funkcja();
  ...
  return 0;
}
```

Wskaźniki do klas

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

class CRectangle {
    int width, height;
  public:
    void set_values (int, int);
    int area (void) {return (width * height);}
};

void CRectangle::set_values (int a, int b) {
  width = a;
  height = b;
}

int main () {
  CRectangle a, *b, *c;
  CRectangle * d = new CRectangle[2];
  b= new CRectangle;
  c= &a;
  a.set_values (1,2);
  b->set_values (3,4);
  d->set_values (5,6);
  d[1].set_values (7,8);
  cout << "a area: " << a.area() << endl;
  cout << "*b area: " << b->area() << endl;
  cout << "*c area: " << c->area() << endl;
  cout << "d[0] area: " << d[0].area() << endl;
  cout << "d[1] area: " << d[1].area() << endl;
  delete[] d;
  delete b;
  return 0;
}
```

Liczby zespolone - przykład klasy

```
class Complex{
  double re, im;
  public:
   void set_values(double, double);
   double modul();
};
void Complex::set_values(double a, double b){
    re = a;
    im = b;
}
double Complex::modul(){
    return (sqrt(re*re + im*im));
}
main(){
  Complex z;
  z.set_values(3,4);
  cout << "Modul wynosi: " << z.modul() << endl;
  return 0;
}
```