

Programowanie

Dariusz Wardecki, wyk. XII

Powtórzenie

Znajdź błąd

```
class Complex{
    double re, im;
public:
    Complex(double, double);
    Complex(double);
    Complex();
};
...
Complex(double a, double b):re(a), im(b) {}
Complex(double a):re(a), im(0) {}
Complex():re(0), im(0) {}

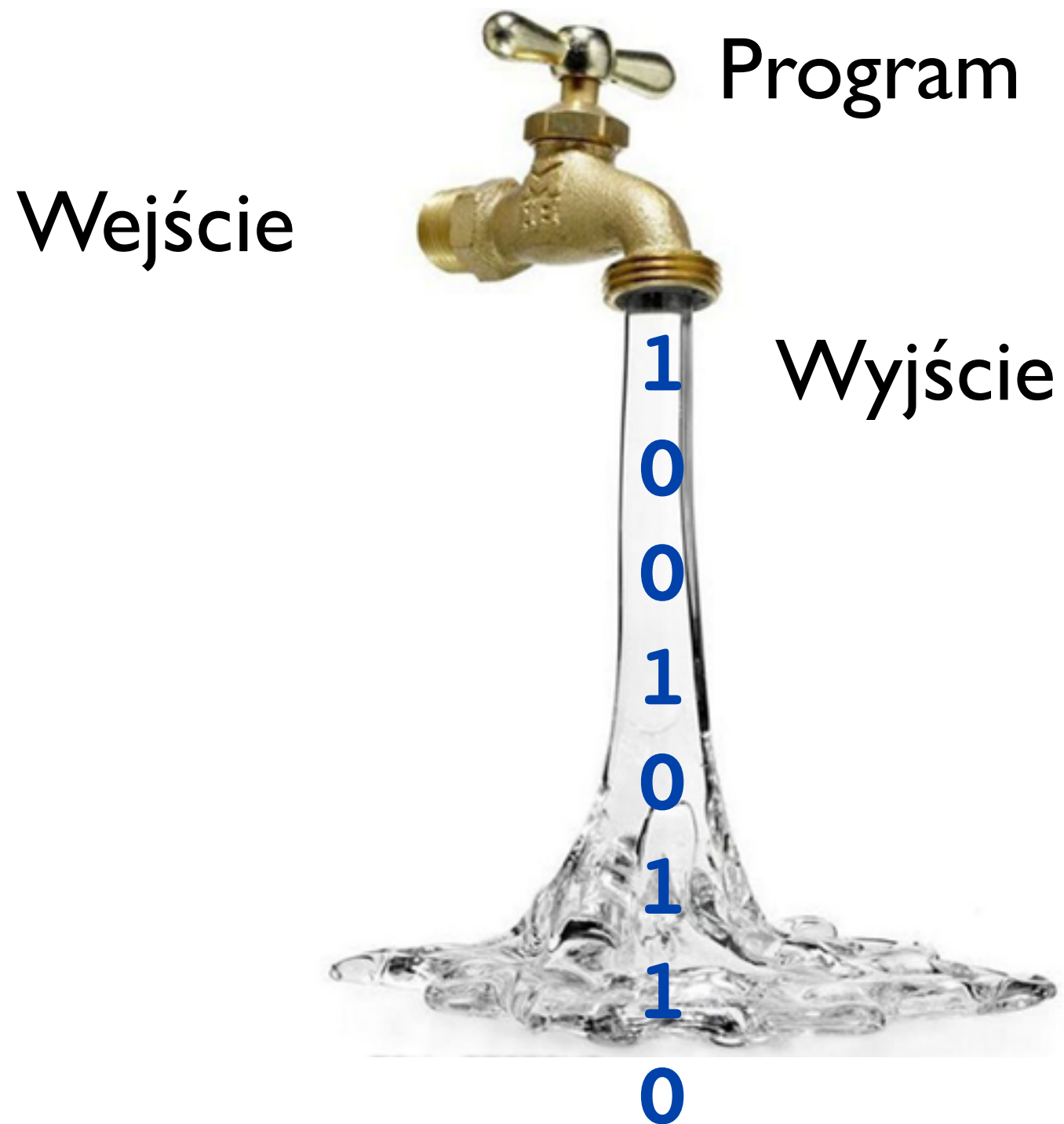
int main(){
    Complex z1(3,4);
    Complex z2(4);
    Complex z3;
    ...
}
```

Powtórzenie

Znajdź błąd

```
...  
class Punkt{  
    public:  
    double x, y;  
    void set(double, double);  
    void set();  
};  
void Punkt::set(double a, double b){  
    x = a;  
    y = b;  
}  
void Punkt::set(){  
    x = 0;  
    y = 0;  
}  
class Odcinek{  
    Punkt p1, p2;  
    public:  
    double dlugosc();  
};  
double Odcinek::dlugosc(){  
    double dl = sqrt((p1.x - p2.x)*(p1.x - p2.x) + (p1.y - p2.y)*(p1.y - p2.y));  
    return (dl);  
}  
int main(){  
    Punkt r0;  
    Punkt r1(2.4, 10.2);  
    Odcinek s(r0, r1);  
    cout << s.dlugosc() << endl;  
    return 0;  
}
```

Strumienie



Strumienie

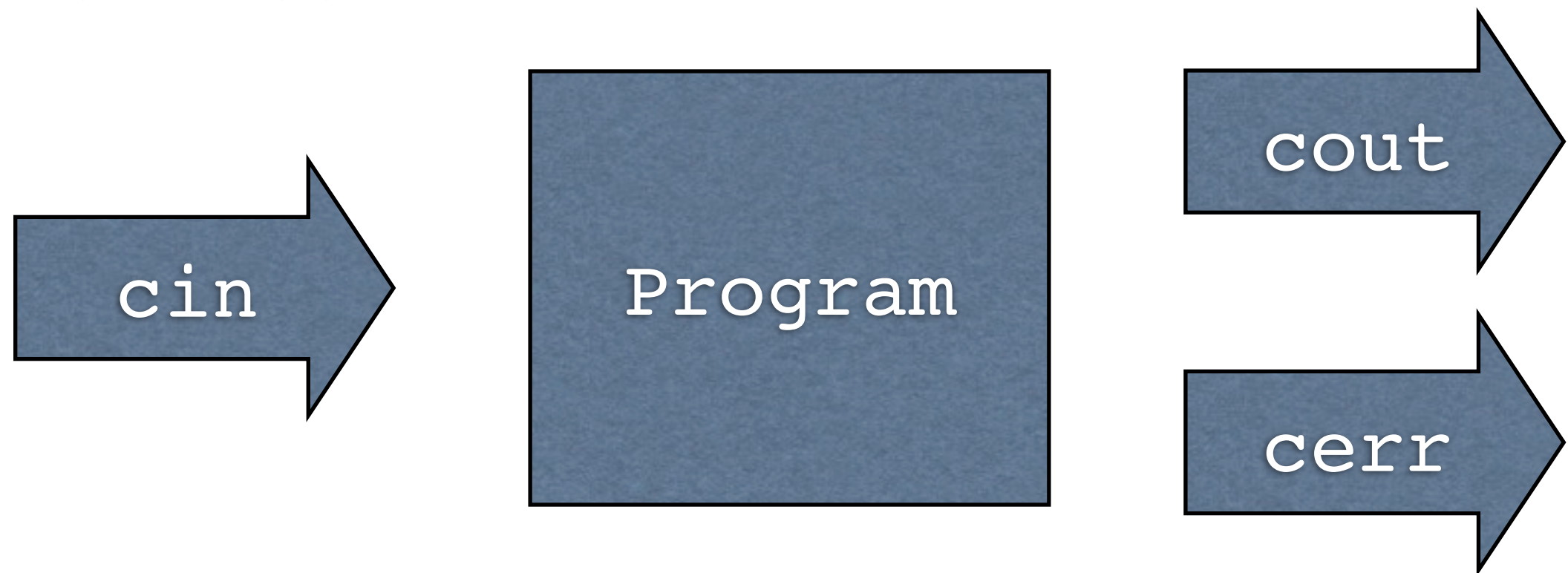
Strumień standardowy

Program



Strumienie

Strumień standardowy
biblioteka:
`<iostream>`



Pobieranie danych ze strumienia

operator lub funkcja	Uwagi
>>	Operator >> domyślnie pomija białe znaki. Wczytuje ciąg znaków do pojawiania się kolejnego białego znaku
<code>get(char* gdzie, int ile, char ogran = '\n')</code>	Ogranicznik nie jest wyjmowany ze strumienia
<code>getline(char* gdzie, int ile, char ogran = '\n')</code>	Ogranicznik jest wyjmowany ze strumienia
<code>read(char* gdzie, int ile)</code>	Nie dopisuje na końcu stringu znaku NULL

Zapisywanie danych do strumienia

operator lub funkcja	Uwagi
<<	przekierowanie do strumienia
<code>put(char znak)</code>	Wstawia do strumienia jeden znak
<code>write(char* skąd, int ile)</code>	Wstawia do strumienia określoną ilość danych

Przykład

```
#include<iostream>
#include<sstream>
#include<string>

using namespace std;

// wypisanie pierwszej kolumny z 2-kol pliku
int main(int argc, char *argv[])
{
    string line;
    double x,y;
    while(getline(cin, line)){
        istringstream stream(line); // zamiana string na strumien

        stream >> x >> y;
        cout << x << endl;
    }
    return 0;
}
```

Przykładowe uruchomienie

```
./prog < input > output
```

Pliki tekstowe - odczyt

```
#include<iostream>
#include<fstream>
```

Dodajemy klasę fstream

```
using namespace std;
```

```
int main(int argc, char *argv[]){
```

Definiujemy bufor na dane

```
char buff[100];
```

Definiujemy i otwieramy
strumień

```
ifstream plik;
```

```
plik.open("dane.txt");
```

```
plik.getline(buff, 99);
```

```
cout << buff << endl;
```

Wczytujemy dane ze
strumienia do bufora

```
...
```

```
...
```

```
plik.close();
```

```
return 0;
```

Zamykamy strumień

```
}
```

Przykład

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main(int argc, char* argv[])
{
    //string bufor;
    char bufor[256];
    ifstream input;
    ofstream output(argv[2]);

    input.open(argv[1]);

    //getline(input, bufor);
    //input.getline(bufor, 256, '<');
    //input.get(bufor, 256, '<');
    //input.read(bufor, 256);
    //input.seekg(4); // ustawianie pozycji w pliku na n-tym bajcie
    input >> bufor;

    output << bufor << endl;
    output.put('%');
    output.write(bufor, 3);
    //input.getline(bufor, 256);
    //cout << bufor << endl;

    input.close();
    output.close();
    return 0;
}
```