Young Programmer

www.inceptio.org.pl

Organizator:

fundacja INCEPTIO

Partnerzy:

Wydział Fizyki Politechnika Warszawska

Stowarzyszenie KLATRAT

Dofinansowanie:

Miasto Stołeczne Warszawa

Zakochaj się w Warszawie

Young Programmer

www.inceptio.org.pl

Zajęcia nr 5

Algorytmy i wskaźniki

dr inż. Łukasz Graczykowski
mgr inż. Leszek Kosarzewski
Wydział Fizyki Politechniki Warszawskiej

Organizator:

Partnerzy:

Dofinansowanie:

MIASTO STOŁECZNE WARSZAWA

Plan

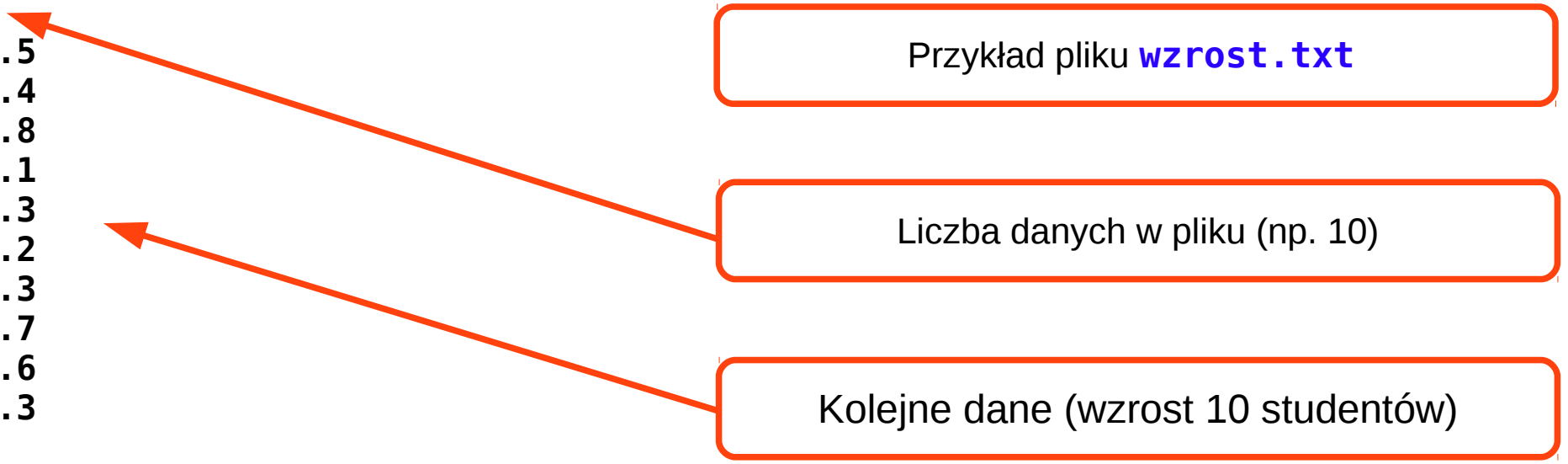
- Zapis i odczyt z plików tekstowych
- O tablicach ciąg dalszy
- Referencje i wskaźniki

Dobra rada

ZAWSZE zaczynajcie pisanie swojego programu od NAJMNIEJSZEJ możliwej kompilującej się wersji.

Odczyt danych z pliku

```
10
175.5
168.4
154.8
181.1
168.3
156.2
174.3
173.7
169.6
175.3
```



Odczyt danych z pliku

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

using namespace std;

int main ()
{

ifstream file;  // mozna tez: ifstream file("wzrost.txt");
file.open("wzrost.txt");

int n = 0;
file>>n;

double a;

for (int i = 0; i < n; i++)
{
   file>>a;
   cout<<"a = "<<a<<endl;
}

file.close();

return 0;

}
```

Używamy biblioteki **<fstream>**
obsługuje strumienie zapisu/odczytu plików

Tworzymy obiekt **ifstream** dla odczytu pliku

Otwieramy plik podając nazwę: **"wzrost.txt"**

W tym przypadku wczytujemy liczbę
znajdująca się na początku pliku

Odczytujemy wzrost w pętli I wypisujemy na ekran

Zamykamy plik

Zapis danych do pliku

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

using namespace std;

int main ()
{

ofstream file;  // można też: ofstream file("liczby.txt");
file.open("liczby.txt");

for (int i = 0; i < 10; ++i)
{
   file<<i<<endl;
}

file.close();

return 0;

}
```

Używamy biblioteki **<fstream>**
obsługuje strumienie zapisu/odczytu plików

Tworzymy obiekt **ofstream** dla zapisu pliku

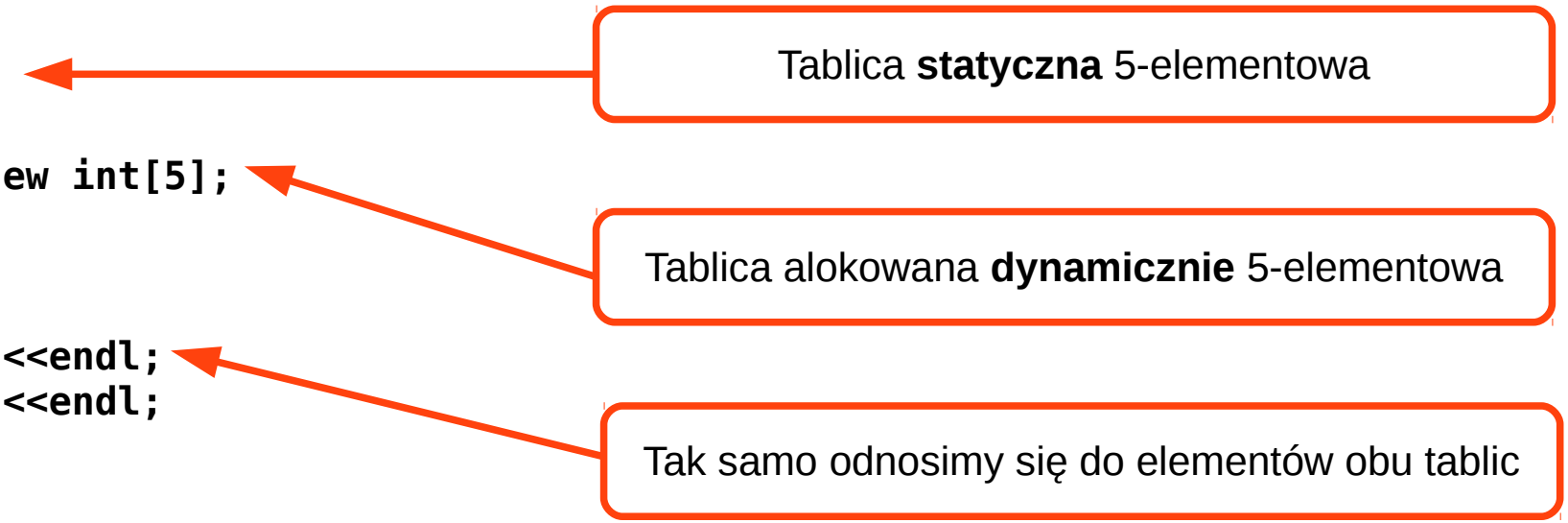
Tworzymy nowy plik podając nazwę:**"liczby.txt"**

Zapisujemy kolejne liczby 0..9 do pliku

Zamykamy plik

Tablice (ciąg dalszy 1)

```
#include <iostream>

using namespace std;

int main ()
{
int tab1[5];

int *tab2 = new int[5];

tab1[0] = 1;
tab2[0] = 2;
cout<<tab1[0]<<endl;
cout<<tab2[0]<<endl;


return 0;

}
```



Tablice (ciąg dalszy 2)

```
#include <iostream>

using namespace std;

int main ()
{
int tab1[5];

int *tab2 = new int[5];

delete[] tab2;
tab2 = new int[10];


return 0;

}
```

Rozmiaru tablicy statycznej **nie da się zmienić**

Usuwamy tablicę i tworzymy nową z innym rozmiarem

Referencje i wskaźniki

```
#include <iostream>

using namespace std;

int main ()
{
int a = 5;
int &b = a;
b = 10;
cout<<a<<endl; //10
cout<<b<<endl; //10

int aa = 5;
int bb = 10;

int *c;
c = &aa
cout<<(*c)<<endl; //5
c = &bb;
cout<<(*c)<<endl; //10

return 0;

}
```

Wskaźniki – klasy 1

```
#include <iostream>

using namespace std;

class Test
{
    int a;
    public:
    Test(){a = 100;}
    ~Test(){cout<<”Destruktor!”<<endl;}
    void Ustaw(int Aa){a = Aa;}
    void Wypisz(){cout<<a<<endl;}
};

int main ()
{
Test obiekt;
obiekt.Wypisz() //100

Test *wskaznik = new Test();
wskaznik->Wypisz(); //100


delete wskaznik; //Destruktor!

return 0;

}
```

Tworzenie obiektu
Użycie konstruktora domyślnego

Tworzenie obiektu “przez wskaźnik”
Użycie konstruktora domyślnego

Usunięcie obiektu z pamięci (użycie destruktora)

Wskaźniki – klasy 2

```
#include <iostream>

using namespace std;

class Test
{
    int a;
    public:
    Test(){a = 100;}
    ~Test(){cout<<”Destruktor”<<endl;}
    void Ustaw(int Aa){a = Aa;}
    void Wypisz(){cout<<a<<endl;}
};

int main ()
{
Test obiekt1;
Test obiekt2;
obiekt2.Ustaw(200);

Test *wskaznik;
wskaznik = &obiekt1;
wskaznik->Wypisz(); //100

wskaznik = &obiekt2;
wskaznik->Wypisz(); //200

return 0;
}
```

“Ustawiamy” wskaźnik na obiekt1

“Ustawiamy” wskaźnik na obiekt2

Young Programmer

www.inceptio.org.pl

Zajęcia nr 5

Algorytmy i wskaźniki

Zadania

Organizator:

fundacja INCEPTIO

Partnerzy:

Wydział Fizyki Politechniki Warszawskiej

Stowarzyszenie KLATRAT

Dofinansowanie:

Miasto Stołeczne Warszawa

Zakochaj się w Warszawie

Zadanie 1: Statystyka

Urząd statystyczny postanowił, w ramach okresowego badania populacji, zmierzyć wzrost pewnej reprezentatywnej grupy studentów. Do opracowania wyników badania niezbędne jest: policzenie średniego wzrostu studentów z badanej próbki, odchylenia standardowego wartości średniej, oraz posortowanie danych.

Plik z danymi ma następujący format:
liczba_studentów
wzrost1
wzrost2
…

Zadania:

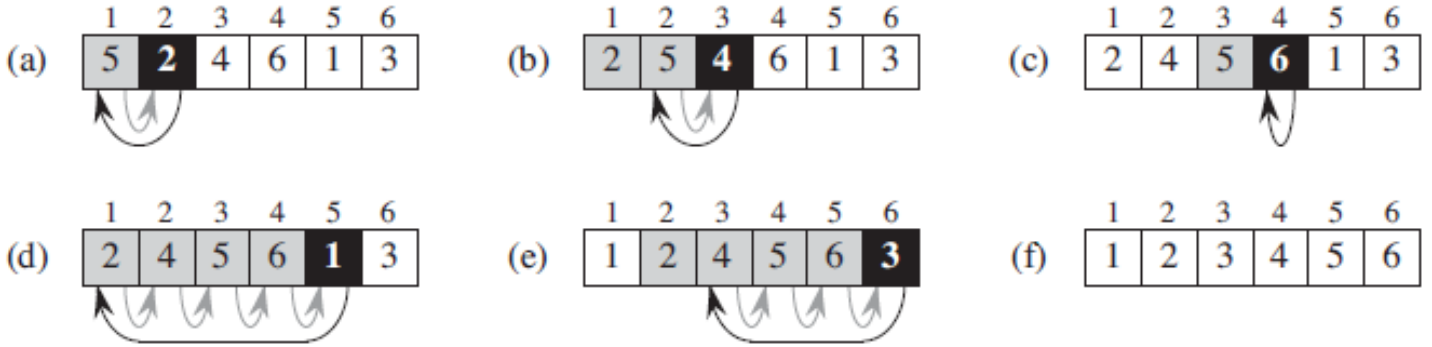
- Stworzyć (dynamicznie) tablicę liczb rzeczywistych (double) o rozmiarze danym liczbą zbadanych studentów i wczytać do niej z zewnętrznego pliku dane.
- Wypisać na ekran tablicę danych – funkcja zewnętrzna:
  void wypisz(**const float*** tab, **int** n)
- Obliczyć i wypisać na ekran wartość średnią wczytanej próbki danych – funkcja zewnętrzna:
  float srednia(**const float*** tab, **int** n)
- Zaimplementować funkcję sortującą dane za pomocą algorytmu sortowania przez wstawianie, tzw. Insert Sort (patrz **Uwaga 1** – następna strona) – funkcja zewnętrzna:
  void sortuj(**float*** tab, **int** n)
- Program powinien działać w pętli **while**, a każda z powyższych opcji powinna być realizowana za pomocą odpowiedniej opcji używając **switch**-**case**.
- Plik z danymi należy ściągnąć z:
  http://www.if.pw.edu.pl/~lgraczyk/PP2015/lab05/wzrost.txt

Zadanie 1: Statystyka c.d.

Uwaga 1:
Schemat działania algorytmu sortowania przez wstawianie (za W*ikipedia*):
https://pl.wikipedia.org/wiki/Sortowanie_przez_wstawianie (zobacz animację!)

1. Utwórz zbiór elementów posortowanych i przenieś do niego dowolny element ze zbioru nieposortowanego.
2. Weź dowolny element ze zbioru nieposortowanego.
3. Wyciągnięty element porównuj z kolejnymi elementami zbioru posortowanego póki nie napotkasz elementu równego lub elementu większego (jeśli chcemy otrzymać ciąg niemalejący) lub nie znajdziemy się na początku/końcu zbioru uporządkowanego.
4. Wyciągnięty element wstaw w miejsce gdzie skończyłeś porównywać.
5. Jeśli zbiór elementów nieuporządkowanych jest niepusty wróć do punkt 2.

https://www.youtube.com/watch?v=ROalU379l3U
https://www.youtube.com/watch?v=DFG-XuyPYUQ
https://en.wikipedia.org/wiki/Insertion_sort



Zadanie 1: Statystyka c.d.

Piszemy **klasę**, która będzie zawierała w sobie tablicę, kontstruktor oraz metody do liczenia średniej oraz sortowania (zadanie projektowe).

Zadanie 2: Wektor

Piszemy klasę do przechowywania nieskończonej ilości liczb typu double.

Zauważmy, że zwykła tablica, statyczna bądź dynamiczna, zawsze ma ograniczoną ilość elementów. Jendym z rozwiązań pozwalających na obejście tego problemu jest tzw. “wektor”.

Wektor jest strukturą danych zawierającą dynamicznie tworzoną tablicę, która działa w ten sposób, że w momencie gdy wstawiany jest ostatni element tablicy, jej rozmiar zwiększa się dwukrotnie. Przykład:

1) Tworzymy nowy obiekt typu wektor → mamy tablicę jednoelementową: [0]
2) Dodajemy liczbę na jedno jedyne miejsce [0] → tablica jest 2x rozsrzerzana: [0][1]
3) Dodajemy liczbę na miejsce [1] → tablica jest 2x rozszerzana: [0][1][2][3]
4) Dodajemy liczbę na miejsce [2] → nie ma potrzeby rozszerzania: [0][1][2][3]
5) Dodajemy liczbę na miejsce [3] → znowu rozszerzamy 2x: [0][1][2][3][4][5][6][7]
6) Dodajemy liczbę na miejsce [4] → nie ma potrzeby rozszerzania: [0][1][2][3][4][5][6][7]
7) Dodajemy liczbę na miejsce [6] → nie ma potrzeby rozszerzania: [0][1][2][3][4][5][6][7]
Itp.

Zadanie 3: Lista (dla zaawansowanych)

Patrz zadanie:
http://www.if.pw.edu.pl/~lgraczyk/JP2015/lab05/plus/zadanie5_10.11.2015.pdf