|  |
| --- |
| **Programowanie Obiektowe** |
| **Autor: Hubert Zając, Elektronika 283056** |
| https://github.com/huuubertz/STL.git |

**Zadanie 1**

**OPIS:**

Wykorzystane funkcje dla:

Vector: size(), resize(), (constructor), back(), front(), push\_back(), pop\_back(), clear()

List: size(), push\_front(), push\_back(), begin(), end(), clear(), iterator()

Do kodu podajemy liczby dodatnie. Gdy podamy ujemną liczbe program kończy swoje działanie i pokazuje nam, jaka liczba spowodowała koniec programu. Działanie kodu jest opisane w komentarzach w poniższym kodzie.

**KOD:**

void sample\_stl\_program(){

// definiujemy liste i vector

std::list<int> lista;

std::vector<float> wektor(10);

// przykładowe dane przekazywane do listy

int liczba;

float suma=0;

// przekazujemy dane dopoki liczba nie będzie mniejsza od 0

std::cout << "Podaj liczby do przefiltrowania" << std::endl;

do{

std::cin >> liczba;

// zapisuj każdą podaną liczbe do listy

if (liczba % 2 == 0){

lista.push\_front(liczba);

}

else lista.push\_back(liczba);

// Jeżeli lista > 10 to zrob sume z tych liczb i przekaz ją do wektora

if (lista.size() >= 10){

std::cout << lista.size() << " danych zostalo wpisanych" << std::endl;

// sumujemy te 10 liczb

// ustawiamy iterator listy na jej początek

std::list<int>::iterator iterator\_listy = lista.begin();

while (iterator\_listy != lista.end()) {

//std::cout << \*iterator\_listy << std::endl;

// sumujemy 10 liczb z listy

suma += \*iterator\_listy;

++iterator\_listy;

}

// wyczyść liste

lista.clear();

// zapisujemy do wektora sume 10 liczb z listy i dzielimy przez 10

wektor.push\_back(suma / 10);

std::cout << wektor.at(1) << std::endl;

// Jeżeli długość wektora będzie równa 10 to zwiększamy jego rozmiar do 20;

if (wektor.size() == 10){

wektor.resize(20);

}

// wypisujemy liczbe na wyjscie

std::cout << wektor.front() << std::endl;

// gdzy wektor ma wielkosc 20 sciagamy z niego dane i sumujemy

if (wektor.size() == 20){

int idx = 0;

suma = 0;

while (idx != wektor.size()){

suma += wektor.back();

wektor.pop\_back();

}

//std::cout << suma << std::endl;

// dodaj do wektora element suma podzielony przez 20

wektor.push\_back(suma / 20);

// wyświetl sume sredniech arytmetycznych po kolejnym uśrednieniu przez 20

std::cout << wektor[wektor.size()-1] << std::endl;

}

}

} while (liczba > 0);

// wyświetl liczbe, która spowodowała wysypanie programu

std::list<int>::iterator iterator\_listy = lista.begin();

std::cout << \*iterator\_listy << std::endl;

}

**WNIOSKI:**

Treścią zadania było, żeby napisać kod korzystający z metod, które daje nam biblioteka Vector i List nie miał on mieć sensu, ani robić nic sensownego. Fajnym dodatkiem zamiast warunku czy liczba z listy jest równa np. 7 jest metoda remove\_if(), do której potrzebna jest wartość bool’owska np. jakaś funkcja która sprawdza nam czy liczba jest parzysta czy nie i zwaracająca wartość true lub false.

**Zadanie 5**

**OPIS:**

**KOD:**

**WNIOSKI:**

**Zadanie 6**

**OPIS:**

**KOD:**

**WNIOSKI:**

**Zadanie 7**

**OPIS:**

**KOD:**

**WNIOSKI:**