**Лабораторна робота № 2**

**Тема:**Робота з лінійними списками. Конструктор і деструктор класу

**Мета:**Навчитись використовувати конструктори і деструктори класів,  створювати класи для опису лінійних списків.

## Порядок виконання роботи

**Завдання 1.**Реалізувати один простий клас згідно варіанту індивідуального завдання, що містить закриті данні, а саме два типа даних: числове та рядкове, реалізоване через покажчик на char. Потрібно створити декілька екземплярів класу статично і динамічно, а також масив та продемонструвати дію всіх конструкторів і методів. Реалізувати методи:

* Scan - ввід даних з клавіатури у поля класу;
* Print  - констатний метод виводу даних на екран;
* Конструктор по замовчуванню;
* Конструктор ініціалізації Клас(сhar\*, int);
* Конструктор копіювання Клас(const Клас&);
* Деструктор
* Методи доступу та закритих данних  Get та Set.

Варіант 7

class Airplane

{

char \*Model;

int Power;

public:

Airplane();

Airplane( char \* , int  );

Airplane(  const Airplane& );

void SetModel( char \* );

char \* GetModel( );

void SetPower( int  );

int  GetPower( );

void Print ();

void Input ();

~Airplane();

};

Код програми

#include <iostream>

#include <string.h>

using namespace std;

class **Airplane**

{

char \*Model;

int Power;

public:

**Airplane**();

**Airplane**( char \* , int );

**Airplane**( const Airplane& );

void **SetModel**( char \* );

char \* **GetModel**( );

void **SetPower**( int );

int **GetPower**( );

void **Print** ();

void **Input** ();

~**Airplane**();

};

Airplane::**Airplane**() {

Model = nullptr;

Power = 0;

}

Airplane::**Airplane**(char \*\_Model, int \_Power) {

Model = \_Model;

Power = \_Power;

}

Airplane::**Airplane**(const Airplane& T) {

Model = new char[40];

for (int i = 0; i < 40; i++){

Model[i] = T.Model[i];

}

Power = T.Power;

}

void Airplane::**SetModel**(char\* \_Model) {

Model = \_Model;

}

char\* Airplane::**GetModel**() {

return Model;

}

void Airplane::**SetPower**(int \_power){

Power=\_power;

}

int Airplane::**GetPower**() {

return Power;

}

void Airplane::**Print**() {

cout << Model << endl << Power << endl;

}

void Airplane::**Input**() {

char a[40];

cin >> a;

Model = new char[40];

for (int i = 0; i < 40; i++){

Model[i] = a[i];

}

cin >>Power;

}

Airplane::~**Airplane**() {

delete Model;

}

int **main**()

{

char str [] = {"SC-123"};

Airplane A, \*B = new Airplane(str, 6);

A.Input();

A.Print();

B->Print();

Airplane C = A;

Airplane mas[] = { A, \*B, C};

for (int i = 0; i < 3; i++) {

mas[i].Print();

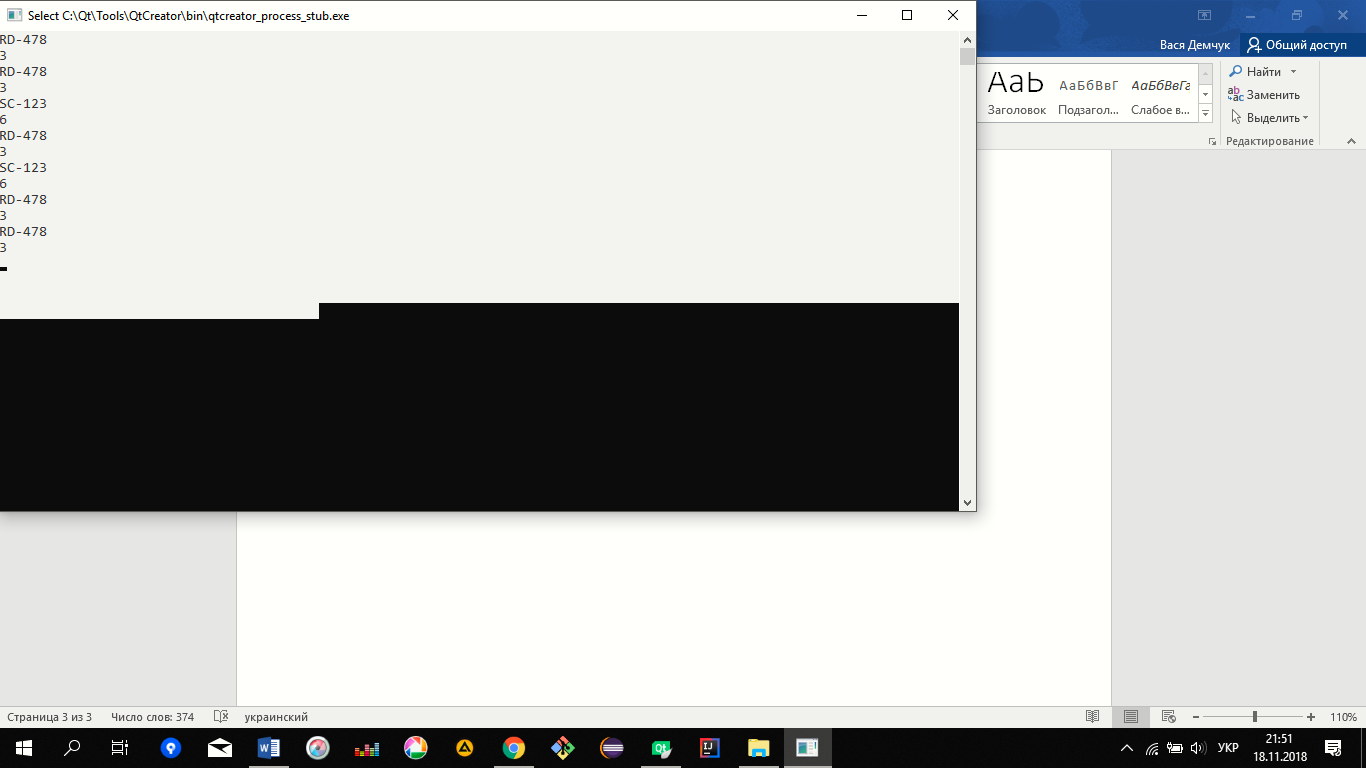
}

C.Print();

return 0;

}

Вивід програми



**Завдання 2.**Виконати завдання згідно варіанту індивідуального завдання:Організувати чергу, заповнити її випадковими числами і зробити функції додавання в чергу та знаходження середнього арифметичного чисел записаних у чергу та їх кількості. Величина черги наперед невідома.

Код програми

#include <iostream>

#include<queue>

using namespace std;

class **Queue**{

private:

queue<float> MyQueue;

public:

**Queue**(){

float ser=0;

for(int i = 0;i<rand();i++){

MyQueue.push(rand());

ser+=MyQueue.back();

}

cout<<"midlenumber-"<<ser/10<<endl;

}

void **addElement**(int x){

MyQueue.push(x);

}

void **deleteElement**(){

MyQueue.pop();

}

void **getFrontAndBack**(){

cout<<MyQueue.front()<<" "<<MyQueue.back()<<endl;

}

void **GetSizeQueue**(){

cout<<MyQueue.size(); }

};

int **main**()

{

Queue a;

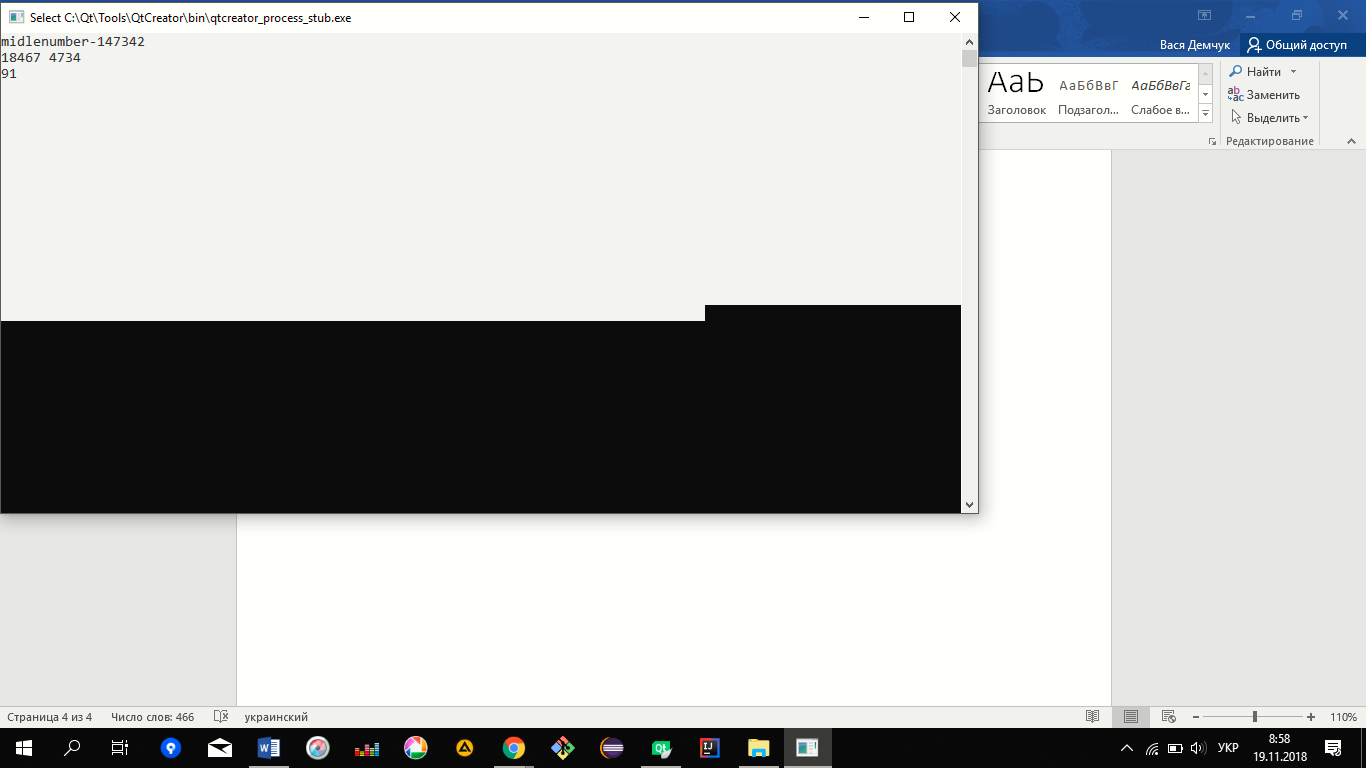
a.getFrontAndBack();

a.GetSizeQueue();

return 0;

}

Вивід програми



**Виисновок:**Навчився використовувати конструктори і деструктори класів,  створювати класи для опису лінійних списків.