Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

ЗВІТ   
про виконання лабораторної роботи №1  
з дисципліни «Об'єктно-орієнтований аналіз та конструювання програмних систем»

Виконав: студент групи TI-92

Черноусов Денис Ігорович

Перевірив: Касьянов Антон Сергійович

Київ - 2020

**16 варіант**

1. Визначте структуру, в якій містилися б тільки числові значення функції f(x) (п.2), параметри a, b, значення кроку dxта граничних значень аргументу функції x1***,*** x2 (п.1)**.**
2. Розробіть програму, що має працювати при різних значеннях параметрів функції, граничних значень x1***,*** x2 і кроку аргументу функції.
3. Створіть функцію ReadPar(), призначення якої зчитування з клавіатури за допомогою функції scanf()значень параметрів функції та параметрів dx, x1***,*** x2.ma
4. Знайдіть значення функції f(x)для всіх точок діапазону [x1,x2], зокрема для випадків, коли dx після можливих розрахунків може перевищувати точне значення на 0,1%.
5. Створіть функцію Tab(), що виводить на екран розраховані значення функції f(x) та значення аргументу x з такою кількістю значущих цифр, як наведено у таблиці (п.3). При створенні таблиці слід застосувати символи псевдографіки.
6. Для розрахованих значень функцій f(x) обчисліть параметри S1, S2, наведені в п.6. Створіть масив дійсних чисел rnd[i] такого ж розміру, що й масив значень функції f(x). Значення масиву дійсних чисел повинні бути випадковими в діапазоні [Smin, Smax], де Sminє меншим, а Smaxє більшим з поміж S1 таS2. Крок випадкових значень має складати \*Sсер, Sсер=(Smin+Smax)/2 **(** визначено вп.5).
7. Виведіть на екран значення S1 таS2 та створеного масиву випадкових чисел в табличному режимі.
8. Визначте лінійну функцію перетворення розрахованих значень функції f(x)в цілі числа діапазону [0, A] (п.5). Створіть функціюPrint(), яка б виводила перетворені значення функції на екран у вигляді вертикальної залежності функції  за допомогою символу, код якого описано в п. 4.
9. Значення параметрів:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 10 | 9 | 0 | 3 | 0.1 |

1. Визначення функції:
2. Вигляд таблиці:

|  |  |  |
| --- | --- | --- |
| *N* |  |  |
| 1 | 0.1 | 1.12345 |
| 2 |  |  |

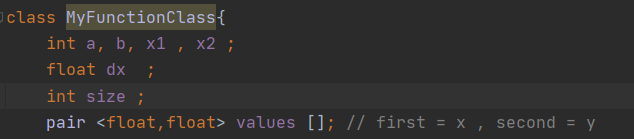
1. Код символу 0xDD {▌}.
2. Параметри: A=55, =0.01.
3. S1 – сума всіх парних за номером точок масиву функції , S2 – максимальне значення функції серед парних розрахованих точок.

Що таке поліморфізм в об'єктно-орієнтованому програмуванні?

Що таке послідовне виконання в структурному програмуванні?

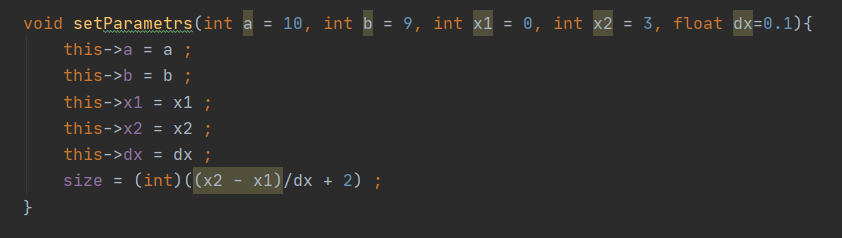
Завдання 1 :

Клас з необхідними полями

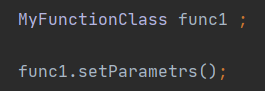


Завдання 2 :

Замість конструктора є метод , який дозволяє встановлювати одному й тому ж об’єкту різні параметри . Є параметри по замовчуванню



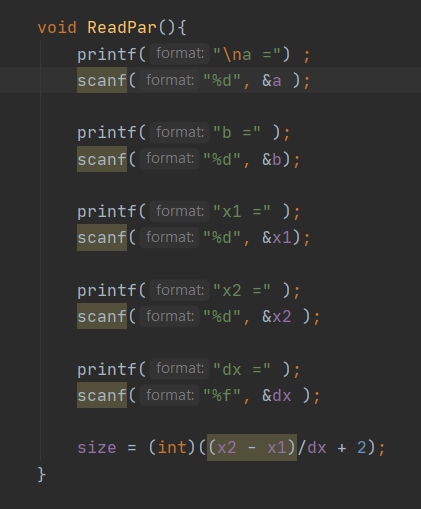
Декларуємо об’єкт і встановлюємо(перевстановлюємо) параметри





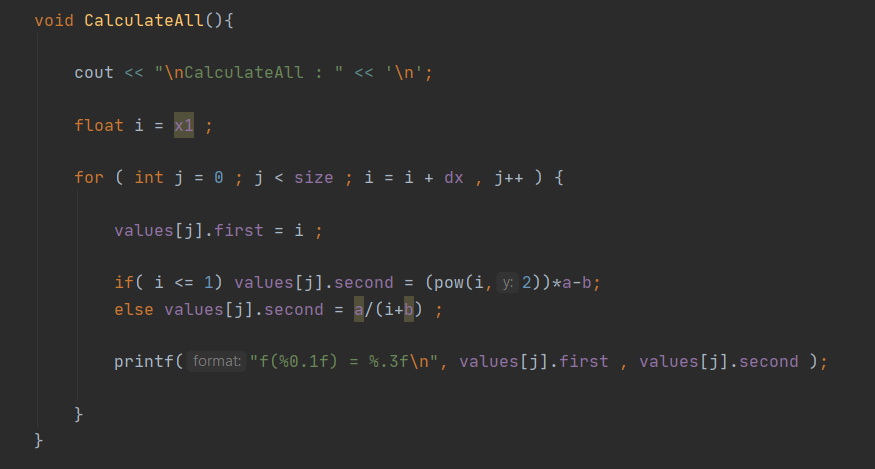
Завдання 3 :

Метод для зчитування ReadPar()

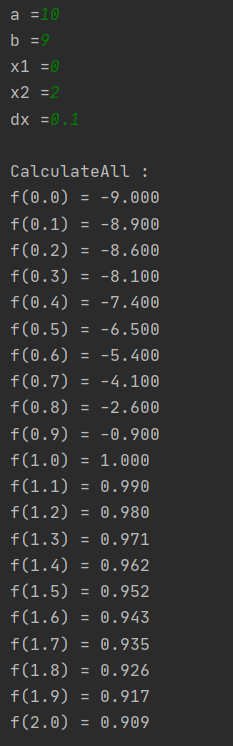


Завдання 4 :

Метод для рахування і виведення всіх значень х і значень функції f(x)

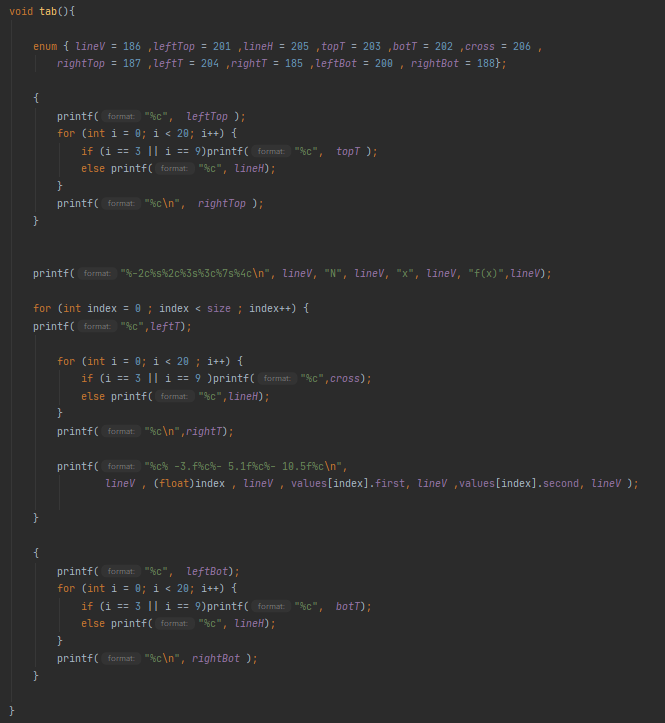


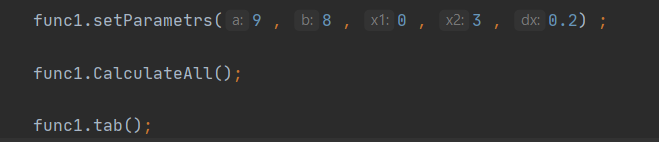
ReadPar() разом з CalculateAll() при використані :



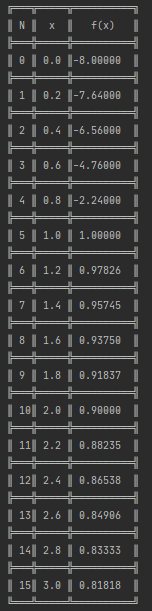
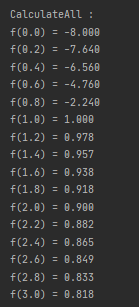
Завдання 5 :

Метод tab() для виведення значень x і значень функції f(x) в табличному вигляді



З ось цим 

Маємо вивід :



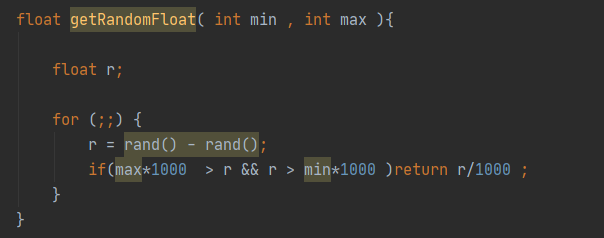
Завдання 6 :

Для створення масиву з випадковими значеннями зробимо метод makeRandomArray() , оновимо значення dx (кроку) як вказано в завдані

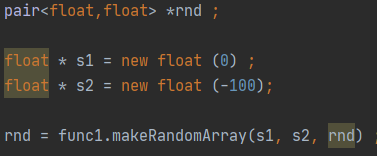


GetRandomFloat() метод для отримання випадкового плаваючого числа ,

яке ми присвоїмо масиву rnd[] за допомогою цикла .

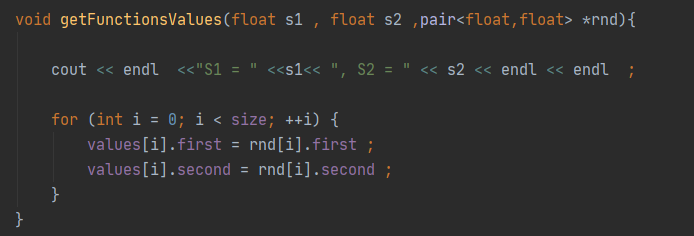


Викликає makeRandomArray() в main’і , перед тим оголосивши декілька вказівків, які нам потрібні для цього методу і методу з наступного завдання

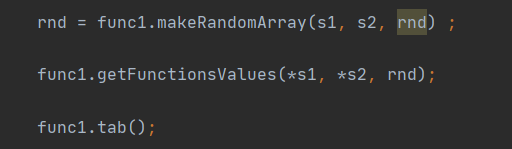


Завдання 7 :

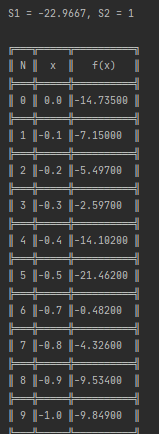
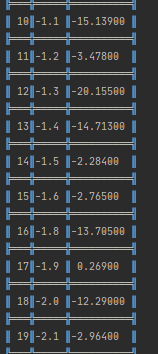
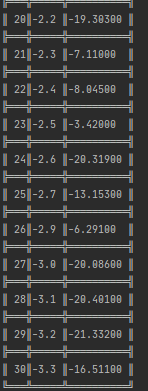
За допомогою метода getFunctionValues() надаємо масиву значень функції values[] нові значення rnd[] і виводимо S1 і S2



Так викликаємо методи в main`і , аби спершу надати значення , а потім вивести таблицю

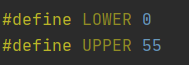


Маємо вивід у вигляді таблички

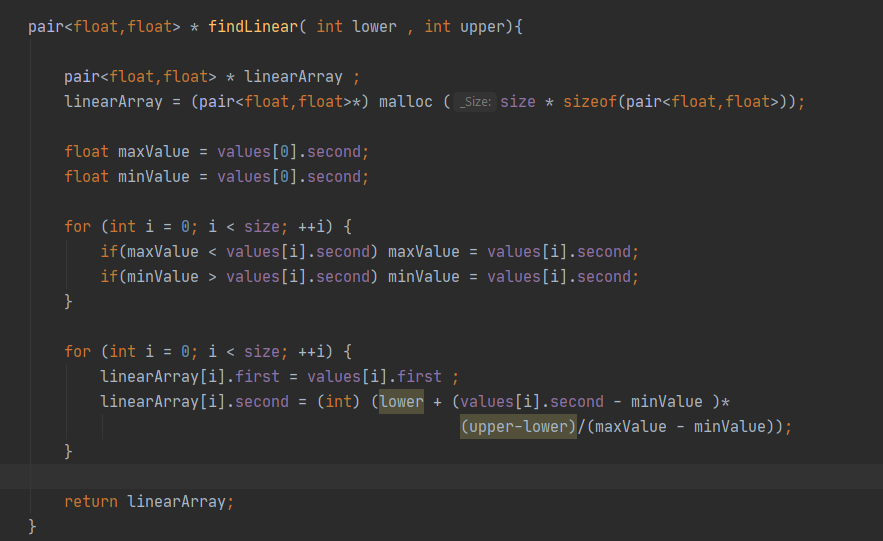


Завдання 8 :

За умовою завдання в нас є межі , тож запишемомо їх через define

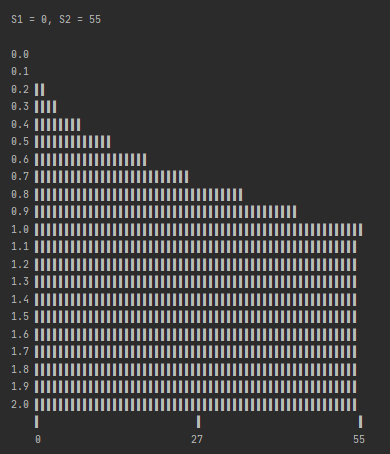
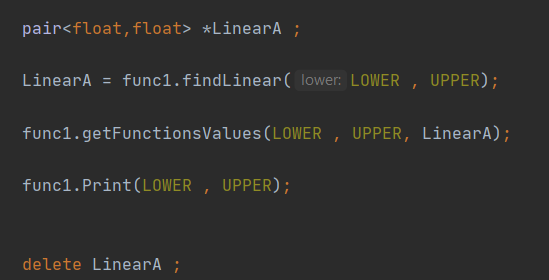


Знайдемо всі значення лінійної функції за допомогою метода findLinear()



Метод Print() для виведення значень у формі графіку



В Main’і використаємо ці методи ось так : вивід метода Print():

Весь код до практичного завдання :

#include <iostream>  
#include <math.h>  
#include <stdlib.h>  
#include <time.h>  
#include <iomanip>  
  
#define LOWER 0  
#define UPPER 55  
  
using namespace std;  
  
class MyFunctionClass{  
 int a, b, x1 , x2 ;  
 float dx ;  
 int size ;  
 pair <float,float> values []; // first = x , second = y  
public:  
  
 void setParametrs(int a = 10, int b = 9, int x1 = 0, int x2 = 3, float dx=0.1){  
 this->a = a ;  
 this->b = b ;  
 this->x1 = x1 ;  
 this->x2 = x2 ;  
 this->dx = dx ;  
 size = (int)((x2 - x1)/dx + 2) ;  
 }  
  
 void CalculateAll(){  
  
 cout << "\nCalculateAll : " << '\n';  
  
 float i = x1 ;  
  
 for ( int j = 0 ; j < size ; i = i + dx , j++ ) {  
  
 values[j].first = i ;  
  
 if( i <= 1) values[j].second = (pow(i,2))\*a-b;  
 else values[j].second = a/(i+b) ;  
  
 printf("f(%0.1f) = %.3f\n", values[j].first , values[j].second );  
  
 }  
 }  
  
 void ReadPar(){  
 printf("\na =") ;  
 scanf("%d", &a );  
  
 printf("b =" );  
 scanf("%d", &b);  
  
 printf("x1 =" );  
 scanf("%d", &x1);  
  
 printf("x2 =" );  
 scanf("%d", &x2 );  
  
 printf("dx =" );  
 scanf("%f", &dx );  
  
 size = (int)((x2 - x1)/dx + 2);  
 }  
  
 void tab(){  
  
 enum { *lineV* = 186 ,*leftTop* = 201 ,*lineH* = 205 ,*topT* = 203 ,*botT* = 202 ,*cross* = 206 ,  
 *rightTop* = 187 ,*leftT* = 204 ,*rightT* = 185 ,*leftBot* = 200 , *rightBot* = 188};  
  
 {  
 printf("%c", *leftTop* );  
 for (int i = 0; i < 20; i++) {  
 if (i == 3 || i == 9)printf("%c", *topT* );  
 else printf("%c", *lineH*);  
 }  
 printf("%c\n", *rightTop* );  
 }  
  
  
 printf("%-2c%s%2c%3s%3c%7s%4c\n", *lineV*, "N", *lineV*, "x", *lineV*, "f(x)",*lineV*);  
  
 for (int index = 0 ; index < size ; index++) {  
 printf("%c",*leftT*);  
  
 for (int i = 0; i < 20 ; i++) {  
 if (i == 3 || i == 9 )printf("%c",*cross*);  
 else printf("%c",*lineH*);  
 }  
 printf("%c\n",*rightT*);  
  
 printf("%c% -3.f%c%- 5.1f%c%- 10.5f%c\n",  
 *lineV* , (float)index , *lineV* , values[index].first, *lineV* ,values[index].second, *lineV* );  
  
 }  
  
 {  
 printf("%c", *leftBot*);  
 for (int i = 0; i < 20; i++) {  
 if (i == 3 || i == 9)printf("%c", *botT*);  
 else printf("%c", *lineH*);  
 }  
 printf("%c\n", *rightBot* );  
 }  
  
 }  
  
 float getRandomFloat( int min , int max ){  
  
 float r;  
  
 for (;;) {  
 r = rand() - rand();  
 if(max\*1000 > r && r > min\*1000 )return r/1000 ;  
 }  
 }  
  
 pair<float,float> \* makeRandomArray(float \*s1 , float \*s2 ,pair<float,float> \*rnd){  
  
 for (int i = 0; i <= size ; i = i + 2) {  
 \*s1 += values[i].second;  
 if(\*s2 < values[i].second) \*s2 = values[i].second ;  
 }  
  
 rnd = (pair<float,float>\*) malloc (size \* sizeof(pair<float,float>));  
  
 int max ;  
 int min ;  
  
 if(\*s1 > \*s2){  
 max = \*s1 ; min = \*s2 ;  
 }else max = \*s2 ; min = \*s1 ;  
  
 srand(time(NULL));  
  
 float x = 0;  
 dx = 0.01 \* ((\*s1+\*s2)/2) ;  
  
 for (int i = 0; i < size ; i++) {  
 rnd[i].first = x ;  
 rnd[i].second = getRandomFloat(min , max) ;  
 x = x + dx ;  
 }  
  
 return rnd ;  
  
 }  
  
 void getFunctionsValues(float s1 , float s2 ,pair<float,float> \*rnd){  
  
 cout << endl <<"S1 = " <<s1<< ", S2 = " << s2 << endl << endl ;  
  
 for (int i = 0; i < size; ++i) {  
 values[i].first = rnd[i].first ;  
 values[i].second = rnd[i].second ;  
 }  
 }  
  
 pair<float,float> \* findLinear( int lower , int upper){  
  
 pair<float,float> \* linearArray ;  
 linearArray = (pair<float,float>\*) malloc (size \* sizeof(pair<float,float>));  
  
 float maxValue = values[0].second;  
 float minValue = values[0].second;  
  
 for (int i = 0; i < size; ++i) {  
 if(maxValue < values[i].second) maxValue = values[i].second;  
 if(minValue > values[i].second) minValue = values[i].second;  
 }  
  
 for (int i = 0; i < size; ++i) {  
 linearArray[i].first = values[i].first ;  
 linearArray[i].second = (int) (lower + (values[i].second - minValue )\*  
 (upper-lower)/(maxValue - minValue));  
 }  
   
 return linearArray;  
 }  
  
 void Print(int lower ,int upper ){  
 for (int i = 0; i < size ; ++i) {  
 printf("%-4.1f",values[i].first ) ;  
 for (int j = 0; j < values[i].second; ++j) {  
 printf("%c" , 0xDD );  
 }  
 printf("\n");  
 }  
  
 cout << setw(5) << (char)0xDD ;  
 cout << setw((upper + lower)/2) << (char)0xDD ;  
 cout << setw((upper + lower)/2) << (char)0xDD << endl ;  
  
 cout << setw(5) << lower ;  
 cout << setw((upper + lower)/2) << (upper + lower)/2 ;  
 cout << setw((upper + lower)/2)<< upper << endl ;  
  
 }  
  
};  
  
int main() {  
  
 MyFunctionClass func1 ;  
  
 func1.setParametrs();  
  
 func1.CalculateAll();  
  
 func1.tab();  
  
 pair<float,float> \*rnd ;  
  
 float \* s1 = new float (0) ;  
 float \* s2 = new float (-100);  
  
 rnd = func1.makeRandomArray(s1, s2, rnd) ;  
  
 func1.getFunctionsValues(\*s1, \*s2, rnd);  
  
 func1.tab();  
  
 delete rnd , s1 , s2 ;  
  
 func1.setParametrs(9 , 8 , 0 , 3 , 0.2) ;  
  
 func1.CalculateAll();  
  
 func1.tab();  
  
 func1.ReadPar();  
  
 func1.CalculateAll();  
  
 func1.tab();  
  
 pair<float,float> \*LinearA ;  
  
 LinearA = func1.findLinear(LOWER , UPPER);  
  
 func1.getFunctionsValues(LOWER , UPPER, LinearA);  
  
 func1.Print(LOWER , UPPER);  
  
  
 delete LinearA ;  
 return 0;  
}

**Теоретичні завдання**

1. Що таке поліморфізм в об'єктно-орієнтованому програмуванні?
2. Що таке послідовне виконання в структурному програмуванні?
   * + 1. Поліморфізм в об'єктно-орієнтованому – це концепція використання єдиного інтерфейсу для взаємодії з даними різних типів. Простіше кажучи, Поліморфізм - це використання різних функцій(методів), які мають одинакову назву, залежно від типів і кількості параметрів цієї функції.
       2. Послідовне виконання – це одна з конструкцій методології структурного програмування. Вона відповідає за послідовне одноразове виконання операторів в програмі .

Висновки : при виконані цієї лабораторної роботи, було пригадано і засвоєно багато матеріалу з 1-го курсу вивчення мови с++ , а саме динамічне виділення пам’яті , об’єктно орієнтоване програмування і маніпуляції з виводом даних.