Министерство образования и науки Российской Федерации

Государственное образовательное учреждение высшего

профессионального образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра информационных технологий и систем

Отчет по лабораторной работе №2

«Решение задач, заданных графически»

по дисциплине

«Алгоритмические языки и программирование»

Разработал

Студент группы 1092

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Ю. Беспалов

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022

Проверил

Преподаватель кафедры ИТИС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Л.И. Винник

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022

Великий Новгород, 2022

**I Постановка задачи**

1. Словесное описание задачи

Программа разрабатывается для того, чтобы по введенным пользователем целым числам A, B и C определить значения переменных X, Y, Z по заданным формулам, а далее в зависимости от области значений, которым принадлежат переменные X и Y, определить значение переменной U по заданным формулам.

Результатом работы программы является сообщение вида «A=…, B=…, C=…, X=…, Y=…, Z=…, U=… Переменная X (не) принадлежит области D1 (D2/ Ни D1, ни D2). Переменная Y (не) принадлежит области D1 (D2/ Ни D1, ни D2). Переменная U вычислялась по формуле …»

Вход: переменные A, B, C.

Выход: значение переменных А, B, C, X, Y, Z, U, сообщение.

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя в программе | Тип | Диапазон | Пр/структура | Вход/Выход/Const |
| Переменные | A  B  C | Целые | [-1000..1000] | простая | вход |
| Переменные | X  Y | Целые | [-1000..1000] | простая | выход |
| Переменные | Z  U | Действит | [-100000..100000] | простая | выход |
| Результат | resultX  resultY  resultU | Строка | Длина 35-56 символов. | простая | выход |

1. Функциональный состав
   1. Организация ввода данных (с контролем диапазона и нецифрового ввода данных).
   2. Обработка данных.
   3. Организация вывода результата.
   4. Интерфейсные задачи.
   5. Многоразовое выполнение программы.
2. Спецификации интерфейса
   1. Организация заставки

Цвет текста – белый, цвет фона – черный, цвет рамки – белый/фиолетовый

|  |
| --- |
| Программа  «lab2»  Беспалов Даниил  Гр. 1092 |
| Esc – выход, ENTER - далее |

* 1. Организация ввода данных

|  |
| --- |
| Введите переменные  [-1000...1000]  A: \_\_  B: \_\_  С: \_\_ |
| ENTER – далее |

* 1. Вывод сообщения об ошибке

|  |
| --- |
| Введите переменные  [-1000..1000]  A: \_\_  B: \_\_  С: \_\_ |
| Ошибка ввода данных!  ENTER – далее |

3.4 Организация вывода данных

|  |
| --- |
| Полученные значения А=\*\*  B=\*\*  C=\*\*  X=\*\* Y=\*\*  Z=\*\*  U=\*\*  Переменная X (не) принадлежит области D1 (D2/ Ни D1, ни D2).  Переменная Y (не) принадлежит области D1 (D2/ Ни D1, ни D2).  Переменная U вычислялась по формуле \*\* |
| Esc – повтор ввода, ENTER - далее |

3.5 Интерфейс завершающего экрана

|  |
| --- |
| Программа  «lab2»  Завершила свою работу |
| Нажмите любую клавишу |

1. Внешние данные тестирования

|  |  |  |
| --- | --- | --- |
| Входные данные | Выход | Реакция программы |
| A=2  B=2  C=1 |  | Программа работает нормально |
| A=2001  B=123  C=142 | сообщение об ошибке, повтор ввода данных | Программа работает нормально |
| A=qweq123  B=23  C=17 | сообщение об ошибке, повтор ввода данных | Программа работает нормально |
| A=123ew  B=1421  C=124 | сообщение об ошибке, повтор ввода данных | Программа работает нормально |
| A=-12  B=14  C=120 |  | Программа работает нормально |
| A=1  B=0  C=-1 |  | Программа работает нормально |

1. Пример работающей программы

При запуске программы появляется окно заставки (**интерфейс 3.1**), где пользователю предлагается нажать клавишу Esc для выхода или Enter для продолжения работы программы.

Если пользователь нажал клавишу Esc, то появляется завершающий экран (**интерфейс 3.5**).

Если пользователь нажал клавишу Enter, то появляется окно ввода данных (**интерфейс 3.2**). Курсор устанавливается в место ввода данных, ожидается ввод данных пользователем. Если пользователь ввел некорректные данные (нецифровой ввод данных, вне диапазона), то появляется сообщение об ошибке (**интерфейс 3.3**). После нажатия клавиши Enter место ввода очищается, ожидается повторный ввод данных. Если данные введены верно, то после нажатия клавиши Enter появляется окно вывода результатов (**интерфейс 3.4**), где пользователю предлагается повторить ввод данных (клавиша Esc) или продолжить работу с программой (клавиша Enter).

Если пользователь нажал клавишу Esc, то появляется окно ввода данных (**интерфейс 3.2**), работа с программой начинается заново.

Если же пользователь нажал клавишу Enter, то появляется интерфейс завершающего экрана (**интерфейс 3.5**). После нажатия любой клавиши происходит выход из программы.

**II Проектирование**

1. Уточненное наименование программы lab2;
2. Уточненные данные программы и пользовательские типы
3. Декомпозиция

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Назначение подпрограммы** | **Имя в программе** | **Вход : тип** | **Выход : тип** | **Тест/не тест** | **Примечание** |
| Рисование интерфейса 3.1 | Interface3\_1 | - | - | Не тест |  |
| Рисование интерфейса 3.2 | Interface3\_2 | - | - | Не тест |  |
| Рисование интерфейса 3.4 | Interface3\_4 | - | - | Не тест |  |
| Рисование интерфейса 3.5 | Interface3\_5 | - | - | Не тест |  |
| Перемещение курсора по координатам | gotoxy | x : int  y : int | - | Не тест |  |
| Вычисление значений переменных X, Y и Z | XYZ | - | X : int  Y : int  Z : float | Тест |  |
| Ввод переменной | Input | a : int  y : int | - | Тест |  |
| Вывод сообщения об ошибке ввода | Error | z : int | - | Тест |  |
| Вывод введенных переменных | Output | - | - | Тест |  |
| Определение значений переменных resultX, resultY и resultU | Oblast | x : int  y : int | resultX : string  resultY : string  resultU : string | Тест |  |

#include <iostream>

#include <windows.h>

#include <conio.h>

using namespace std;

void gotoxy(int x, int y)

{

static HANDLE h = NULL;

if (!h)

h = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD c = { x, y };

SetConsoleCursorPosition(h, c);

}

int flag = 1;

int A, B, C, X, Y;

float Z, U;

string resultX, resultY, resultU;

const int x = 61;

const int y = 17;

int mas[y][x] = { { 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2 },

{ 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 },

};

void Error(int z)

{

gotoxy(21,14);

cout << "Ошибка ввода данных!";

gotoxy(28, z);

cout << " ";

}

void Interface3\_1()

{

system("cls");

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

if (mas[i][j] == 1)

cout << "-";

else if (mas[i][j] == 0)

cout << " ";

else if (mas[i][j] == 2)

cout << "|";

}

cout << endl;

}

gotoxy(26, 5);

cout << "Программа";

gotoxy(28, 6);

cout << "'lab2'";

gotoxy(23, 7);

cout << "Беспалов Даниил";

gotoxy(27, 8);

cout << "Гр. 1092";

gotoxy(18, 15);

cout << "Esc – выход, ENTER - далее";

}

void Interface3\_2()

{

flag = 2;

system("cls");

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

if (mas[i][j] == 1)

cout << "-";

else if (mas[i][j] == 0)

cout << " ";

else if (mas[i][j] == 2)

cout << "|";

}

cout << endl;

}

gotoxy(22, 4);

cout << "Ввeдите переменные";

gotoxy(26, 5);

cout << "[-1000...1000]";

gotoxy(26, 6);

cout << "A: ";

gotoxy(26, 7);

cout << "B: ";

gotoxy(26, 8);

cout << "C: ";

gotoxy(24, 15);

cout << "ENTER - далее";

}

void Interface3\_4()

{

flag = 4;

system("cls");

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

if (mas[i][j] == 1)

cout << "-";

else if (mas[i][j] == 0)

cout << " ";

else if (mas[i][j] == 2)

cout << "|";

}

cout << endl;

}

gotoxy(22, 2);

cout << "Полученные значения";

gotoxy(26, 3);

cout << "A=";

gotoxy(26, 4);

cout << "B=";

gotoxy(26, 5);

cout << "C=";

gotoxy(26, 6);

cout << "X=";

gotoxy(26, 7);

cout << "Y=";

gotoxy(26, 8);

cout << "Z=";

gotoxy(26, 9);

cout << "U=";

gotoxy(14, 15);

cout << "Esc - повтор ввода, ENTER - далее";

}

void Interface3\_5()

{

flag = 5;

system("cls");

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

if (mas[i][j] == 1)

cout << "-";

else if (mas[i][j] == 0)

cout << " ";

else if (mas[i][j] == 2)

cout << "|";

}

cout << endl;

}

gotoxy(26, 6);

cout << "Программа";

gotoxy(28, 7);

cout << "'lab2'";

gotoxy(20, 8);

cout << "Завершила свою работу";

gotoxy(20, 15);

cout << "Нажмите любую клавишу";

gotoxy(1, 18);

}

void Input(int\* a, int z)

{

gotoxy(29, z);

while (!(cin >> \*a) || (cin.peek() != '\n') || (\*a < -1000) || (\*a > 1000))

{

cin.clear();

while (cin.get() != '\n');

Error(z);

gotoxy(29, z);

}

}

void XYZ()

{

X = (max(max(A, B), C));

Y = (min(min(A, B), C));

Z = (float)(A + B + C) / 2;

}

void Oblast(int x, int y)

{

int fl = 0;

if (((x >= 0) && (x <= 2)) && ((y <= 1) && (y >= 0)))

{

resultX = "Переменная X принадлежит области D1";

resultY = "Переменная Y принадлежит области D1";

resultU = "Значение U определялось по формуле U=X\*X+Y\*Y";

U = pow(X, 2) + pow(Y, 2);

gotoxy(13, 10);

cout << resultX;

gotoxy(13, 11);

cout << resultY;

gotoxy(7, 12);

cout << resultU;

fl = 1;

}

if (((pow(x, 2) + pow(y + 1, 2)) <= 1) && (y <= -1))

{

resultX = "Переменная Х принадлежит области D2";

resultY = "Переменная Y принадлежит области D2";

resultU = "Значение U определялось по формуле U=X\*Y\*Z";

if (X == 0 || Y == 0 || Z == 0)

{

U = 0;

}

else U = X \* Y \* Z;

gotoxy(13, 10);

cout << resultX;

gotoxy(13, 11);

cout << resultY;

gotoxy(9, 12);

cout << resultU;

fl = 1;

}

if (fl==0)

{

resultX = "Переменная Х не принадлежит ни области D1, ни области D2";

resultY = "Переменная Y не принадлежит ни области D1, ни области D2";

resultU = "Значение U определялось по формуле U=Z";

U = Z;

gotoxy(3, 10);

cout << resultX;

gotoxy(3, 11);

cout << resultY;

gotoxy(11, 12);

cout << resultU;

}

}

void Output()

{

gotoxy(29, 3);

cout << A;

gotoxy(29, 4);

cout << B;

gotoxy(29, 5);

cout << C;

gotoxy(29, 6);

cout << X;

gotoxy(29, 7);

cout << Y;

gotoxy(29, 8);

cout << Z;

gotoxy(29, 9);

cout << U;

gotoxy(47,15);

}

int main()

{

setlocale(LC\_ALL, "ru");

char keys;

do

{

if (flag == 1)

{

Interface3\_1();

do

{

keys = \_getch();

{

if (keys == 13)

{

Interface3\_2();

}

if (keys == 27)

{

Interface3\_5();

}

}

} while (flag == 1);

}

if (flag == 2)

{

Input(&A,6);

Input(&B,7);

Input(&C,8);

XYZ();

gotoxy(37, 15);

do

{

keys = \_getch();

if (keys == 13)

{

Interface3\_4();

}

} while (flag == 2);

}

if (flag == 4)

{

Oblast(X, Y);

Output();

do

{

keys = \_getch();

if (keys == 13)

{

Interface3\_5();

}

if (keys == 27)

{

Interface3\_2();

}

} while (flag == 4);

}

} while (flag != 5);

}