Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №1**

Дисциплина: Основы алгоритмизации и программирования

Тема: “Динамические структуры данных”

Вариант 22

Выполнил:

Студент гр. ИВТ-20-2б

Чувашев Максим Алексеевич

Проверил:

Доцент кафедры ИТАС  
Викентьева О.Л.

Пермь, 2021

**Цель работы**

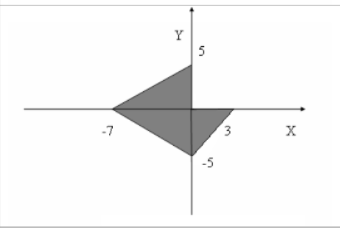
Получить практические навыки работы с однонаправленными списками;

**Постановка задачи**

Задача 1:

1. n++\*m
2. n++<m
3. --m>n

Задача 2:

1. 

Задача 3:

1. Для задачи 1 определить тип заданных выражений и найти их значения.
2. Составить систему тестов и вычислить полученное выражение для нескольких значений Х, определить при каких Х выражение не может быть вычислено.
3. Для задачи 2 записать выражение, зависящее от координат точки X1 и Y1 и принимающее значение TRUE, если точка принадлежит заштрихованной области, и FALSE, если не принадлежит.
4. Составить систему тестов и вычислить полученное выражение для нескольких точек, принадлежащих и не принадлежащих заштрихованной области.
5. Для задачи 3 вычислить значение выражения, используя различные вещественные типы данных (float и double).
6. Результаты всех вычислений вывести на печать.
7. Объяснить полученные результаты.

**Анализ задачи**

1. Задача 1
   1. Для решения 1 задачи необходимо завести 3 переменных n, m, x. Эти переменные необходимы для хранения данных, введенных пользователем. Переменная buf.

int n, m;

double x;

string buf;

* 1. Программа запрашивает у пользователя ввод переменной n и пытается преобразовать полученную строку buf в тип int32. Если пользователь ввел не число tryparse выдаст false, то есть ему не удалось преобразовать данную строку в целочисленное число.

Console.Write("n = ");

buf = Console.ReadLine();

ok = Int32.TryParse(buf, out n);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для n, n - целочисленное!");

}

* 1. Программа будет запрашивать ввод n до тех пор, пока он не введет корректное значение n. То есть целочисленное число.

do

{

Console.Write("n = ");

buf = Console.ReadLine();

ok = Int32.TryParse(buf, out n);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для n, n - целочисленное!");

}

}

while (!ok);

* 1. Аналогичный ввод происходит и для m

do

{

Console.Write("m = ");

buf = Console.ReadLine();

ok = Int32.TryParse(buf, out m);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для m, m - целочисленное!");

}

}

while (!ok);

* 1. Для X так же производятся аналогичные действия, только в условии выхода из цикла добавляется еще одно условие x < - 1. Так как при таких значении решений нет.

do

{

Console.Write("x = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out x);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для x!");

}

else if (x < -1)

{

Console.WriteLine("Вещественное число должно быть больше -1");

}

}

while (!ok || x < -1);

* 1. Далее еще раз выводятся все введенные значения переменных. А также высчитываются требуемые значения из заданий.
     1. В результате 1 примера происходит умножение n \* m, далее инкрементируется n. После этого мы декрементируем n, для того чтобы вернуть изначальное значение n для дальнейших действий.
     2. В результате 2 примера происходит сравнение n и m. Далее n инкрементируется. После выполнения действий декрементируем n для дальнейших действий.
     3. В результате 3 примера происходит сравнение m и n, далее декрементируется m. После выполнения действия инкрементируем m для дальнейших действий.
     4. В результате 4 примера происходит вычисление по математической формуле числа.

Console.WriteLine($"n = {n}, m = {m}, x = {x}");

int result\_1 = n++ \* m;

n--;

bool result\_2 = n++ < m;

n--;

bool result\_3 = --m > n;

m++;

double result\_4 = Math.Pow(2, -x) \* Math.Sqrt(x + Math.Pow(Math.Abs(x), 1 / 4));

* 1. Выводятся все полученные значения в строку.

Console.WriteLine($"n++\*m = {result\_1}, n++<m = {result\_2}, --m>n = {result\_3}, result\_4 = {result\_4}");

1. Задача 2
   1. Для решения 2 задачи необходимо завести 2 переменных X, Y. Эти переменные необходимы для хранения данных, введенных пользователем. Переменная buf.

string buf;

bool ok;

double X, Y;

* 1. Программа запрашивает у пользователя ввод переменной X. Далее программа пытается преобразовать полученную от пользователя строку в вещественное число с помощью tryparse. Если преобразование не удалось, пользователь увидит сообщение об ошибке.

Console.Write("X = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out X);

if (!ok)

{

Console.WriteLine("Введено не число!");

}

* 1. Программа будет запрашивать ввод переменной X до тех пор, пока не сможет преобразовать полученную от пользователя строку в вещественное число.

do

{

Console.Write("X = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out X);

if (!ok)

{

Console.WriteLine("Введено не число!");

}

}

while (!ok);

* 1. Аналогичные действия происходят и с вводом переменной Y

do

{

Console.Write("Y = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out Y);

if (!ok)

{

Console.WriteLine("Введено не число!");

}

}

while (!ok);

* 1. Далее программа выводит в консоль введенные пользователем данные.

Console.WriteLine($"X = {X}, Y = {Y}");

* 1. Так как фигура состоит из 3 прямых и ограничения в виде отсутствия первой четверти координатной плоскости. Для того чтобы понять попадают ли введенные X и Y от пользователя в данный диапазон составляем уравнения всех прямых:
  2. Данные уравнения необходимы так как они являются ограничителями. При X < 0 и X >= -7, Y <= и Y >=. А для X > 0 и X <=3, Y <= 0 и Y >=. Для решения используем bool вместо условного оператора. И выводим получившееся значение в консоль.

double MaxY1, MinY1, MaxY2, MinY2;

bool f1;

MaxY1 = (5 / 7 \* X + 5);

MinY1= (-5 / 7 \* X - 5);

MaxY2 = 0;

MinY2 = 5 / 3 \* X - 5;

f1 = (X < 0 && X >= -7 && Y <= MaxY1 && Y >= MinY1) || (X >= 0 && X <= 3 && Y <= MaxY2 && Y >= MinY2) || ( X == 0 && Y == 0);

Console.WriteLine(f1);

1. 3 задача
   1. Для решения задачи было использовано множество значений типа double и float
   2. Изначально в программе инициализируются значения a и b

Console.WriteLine("a = 100, b = 0.001");

double a = 100;

double b = 0.001;

* 1. Далее по действиям выполняются вычисления для типа double, а результат выводится в консоль

double c = Math.Pow(a - b, 3);

double d = Math.Pow(a, 3);

double e = Math.Pow(b, 2);

double f = 3 \* a \* e;

double g = Math.Pow(b, 3);

double h = Math.Pow(a, 2);

double i = 3 \* h \* b;

Console.WriteLine((c - d) / (f - g - i));

* 1. Для переменных типа float производятся аналогичные действия только результат большинства математических действий придется приводить к типу float. Результат так же выводится на консоль. Разница с предыдущим результатом будет в точности мантиссы.

float c1 = (float)Math.Pow(a - b, 3);

float d1 = (float)Math.Pow(a, 3);

float e1 = (float)Math.Pow(b, 2);

float f1 = (float)3 \* (float)a \* e1;

float g1 = (float)Math.Pow(b, 3);

float h1 = (float)Math.Pow(a, 2);

float i1 = (float)3 \* h1 \* (float)b;

Console.WriteLine((c1 - d1) / (f1 - g1 - i1));

**Код программы 1**

using System;

namespace lab\_1

{

class Program

{

static void Main(string[] args)

{

int n, m;

double x;

string buf;

bool ok = false;

do

{

Console.Write("n = ");

buf = Console.ReadLine();

ok = Int32.TryParse(buf, out n);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для n, n - целочисленное!");

}

}

while (!ok);

do

{

Console.Write("m = ");

buf = Console.ReadLine();

ok = Int32.TryParse(buf, out m);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для m, m - целочисленное!");

}

}

while (!ok);

do

{

Console.Write("x = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out x);

if (!ok)

{

Console.WriteLine("Введено некорректное значение для x!");

}

else if (x < -1)

{

Console.WriteLine("Вещественное число должно быть больше -1");

}

}

while (!ok || x < -1);

Console.WriteLine($"n = {n}, m = {m}, x = {x}");

int result\_1 = n++ \* m;

n--;

bool result\_2 = n++ < m;

n--;

bool result\_3 = --m > n;

m++;

double result\_4 = Math.Pow(2, -x) \* Math.Sqrt(x + Math.Pow(Math.Abs(x), 1 / 4));

Console.WriteLine($"n++\*m = {result\_1}, n++<m = {result\_2}, --m>n = {result\_3}, result\_4 = {result\_4}");

}

}

}

**Код программы 2**

using System;

namespace part\_2

{

class Program

{

static void Main(string[] args)

{

string buf;

bool ok;

Console.WriteLine("Решение 2 задачи");

double X, Y;

do

{

Console.Write("X = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out X);

if (!ok)

{

Console.WriteLine("Введено не число!");

}

}

while (!ok);

do

{

Console.Write("Y = ");

buf = Console.ReadLine();

ok = double.TryParse(buf, out Y);

if (!ok)

{

Console.WriteLine("Введено не число!");

}

}

while (!ok);

Console.WriteLine($"X = {X}, Y = {Y}");

double MaxY1, MinY1, MaxY2, MinY2;

bool f1;

MaxY1 = (5 / 7 \* X + 5);

MinY1= (-5 / 7 \* X - 5);

MaxY2 = 0;

MinY2 = 5 / 3 \* X - 5;

f1 = (X < 0 && X >= -7 && Y <= MaxY1 && Y >= MinY1) || (X >= 0 && X <= 3 && Y <= MaxY2 && Y >= MinY2) || ( X == 0 && Y == 0);

Console.WriteLine(f1);

}

}

}

**Код программы 3**

using System;

namespace part\_3

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Решение 3 задачи");

Console.WriteLine("a = 100, b = 0.001");

double a = 100;

double b = 0.001;

double c = Math.Pow(a - b, 3);

double d = Math.Pow(a, 3);

double e = Math.Pow(b, 2);

double f = 3 \* a \* e;

double g = Math.Pow(b, 3);

double h = Math.Pow(a, 2);

double i = 3 \* h \* b;

Console.WriteLine((c - d) / (f - g - i));

float c1 = (float)Math.Pow(a - b, 3);

float d1 = (float)Math.Pow(a, 3);

float e1 = (float)Math.Pow(b, 2);

float f1 = (float)3 \* (float)a \* e1;

float g1 = (float)Math.Pow(b, 3);

float h1 = (float)Math.Pow(a, 2);

float i1 = (float)3 \* h1 \* (float)b;

Console.WriteLine((c1 - d1) / (f1 - g1 - i1));

}

}

}

**Блок схема 1 задачи**





**Блок схема 2 задачи**



**Блок схема 3 задачи**



Тестирование

1. Задание 1
   1. Тестовые данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Теста | Входные данные | Ожидаемый результат | Полученный результат | Примечания |
| 1 | n = 1 | r1 = 2 | "+" | нет |
| m = 2 | r2 = true |
|  | r3 = false |
| x = 3 | r4 = 0.25 |
| 2 | n = ? | Ошибка ввода | "+" | Программа не дает ввести не число, выводит сообщение об ошибке и просит ввести число заново. |
| m = % | Ошибка ввода |
| x = $ | Ошибка ввода |
| Ошибка ввода |
| 3 | n = 0.5 | Ошибка ввода | "+" | программа не дает ввести  вещественное число в  поля типа int n, m. |
| m = -1.5 | Ошибка ввода |
|  | Ошибка ввода |
| x = -15,5 | Ошибка ввода | А так же не дает присвоить  отрицательное значение x,  так как x > -1. |
| 4 | n = -2 | r1 = -2 | "+" | программа не дает ввести  вещественное число в  поля типа int n, m. |
| m = 1 | r2 = true |
|  | r3 = false |
| x = 1,5 | r4 ≈0,559017 | А так же не дает присвоить  отрицательное значение x,  так как x > -1. |

* 1. Черный ящик

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Переменная | Тип | Т1 | Т2 | Т3 | Т4 |
| n | char |  | + |  |  |
| double |  |  | + |  |
| int | + |  |  | + |
| > 0 | + |  | + |  |
| < 0 |  |  |  | + |
| 0 |  |  |  |  |
| m | char |  | + |  |  |
| double |  |  | + |  |
| int | + |  |  | + |
| > 0 | + |  |  | + |
| < 0 |  |  | + |  |
| 0 |  |  |  |  |
| X | char |  | + |  |  |
| double | + |  | + | + |
| int |  |  |  |  |
| > 0 | + |  |  | + |
| < 0 |  |  | + |  |
| 0 |  |  |  |  |
| **Выходные данные** | | | | | |
| Сообщение об ошибке | |  | + | + |  |
| Число |  | + |  |  | + |

* 1. Белый ящик

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Т1 | Т2 | Т3 | Т4 |
| для n do{}while(!ok) | 0 |  |  |  |  |
| 1 | + |  |  | + |
| n |  | + | + |  |
| для m do{}while(!ok) | 0 |  |  |  |  |
| 1 | + |  |  | + |
| n |  | + | + |  |
| для X while (!ok || x < -1) | 0 |  |  |  |  |
| 1 | + |  |  | + |
| n |  | + | + |  |
| для n if (!ok) | + | + |  |  | + |
| - |  | + | + |  |
| для m if (!ok) | + | + |  |  | + |
| - |  | + | + |  |
| для x if (!ok) | + | + |  |  | + |
| - |  | + | + |  |

1. Задание 2
   1. Тестовые данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Теста | Входные данные | Ожидаемый результат | Полученный результат | Примечания |
| 1 | x = 0 | True | "+" |  |
| y = 0 |
| 2 | x = &(2) | Ошибка ввода | "+" | При попытке ввода недопустимых символов  программа запросит повторный ввод данных |
| y = $(-2) |
| 3 | x = 1 | False | "+" |  |
| y = 1 |
| 4 | x = 1,5 | True | "+" |  |
| y = -2,2 |
| 5 | x = -1,6 | True | "+" |  |
| y = -2,3 |

* 1. Черный ящик

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Переменная | Тип | Т1 | Т2 | Т3 | Т4 |
| x | char |  | + |  |  |
| double | + |  | + | + |
| > 0 |  |  | + |  |
| < 0 |  |  |  | + |
| 0 | + |  |  |  |
| y | char |  | + |  |  |
| double |  |  | + | + |
| > 0 |  |  | + |  |
| < 0 |  |  |  | + |
| 0 | + |  |  |  |
| **Выходные данные** | | | | | |
| Сообщение об ошибке | |  | + |  |  |
| Значение | | + |  | + | + |

* 1. Белый ящик

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Т1 | Т2 | Т3 | Т4 | Т5 |
| для X do{}while(!ok) | 0 |  |  |  |  |  |
| 1 | + |  | + | + | + |
| n |  | + |  |  |  |
| для Y do{}while(!ok) | 0 |  |  |  |  |  |
| 1 | + |  | + | + | + |
| n |  | + |  |  |  |
| для X if (!ok) | + | + |  | + | + | + |
| - |  | + |  |  |  |
| для Y if (!ok) | + | + |  | + | + | + |
| - |  | + |  |  |  |