Тимофеева Наталья

М8О-202Б-19

ООП C#

Лабораторная работа № 1

**Описание работы**

Пункты 1, 2, 3 выполняются в одном проекте.

Пункт  4, посвящен реализации категории агрегация по ссылке и выполняются в отдельном проекте.

1. Изучите понятие абстрагирование. Реализуйте в консольном  проекте функции возвращающие значение и не возвращающие значение (результаты выполнения функций должны выводиться на экран).

2. Дайте определение класса,  в котором эти функции стали членами класса, например, функции распечатывают скрытую строку класса (string str),  которой при инициализации объекта класса в конструкторе присваивается значение, например, (“Hi”).

3. Создайте три объекта этого класса с разными инициализируемыми значениями приветствий, например, “Hello”, “Good morning”. Функции класса должны выводить на экран эти значения, при применении к соответствующему объекту.

 4. Изучите категорию агрегация по ссылке. Представить граф древовидной структуры как иерархию “Целое-Часть”, используя категорию агрегация по ссылке реализуйте граф программно на С#, каждая вершина графа объект соответствующего класса*.*Пусть объект А – целое, а все остальные объекты его части в соответствии с иерархией графа, поставьте правильно стрелки на графе.

В главной программе должно быть реализовано движение по графу от целого к частям.

5. Оформите работу. На графе укажите вид категорий. Сохраните результаты лабораторной.

**Программа 1 (пункты 1, 2, 3)**

using System;

namespace Lab1\_1 {

class Function {

public static int GetMin(int[] array) { //ищет минимальное число в массиве

int min = array[0];

for (int i = 1; i < array.Length; ++i) {

if (array[i] < min)

min = array[i];

}

return min;

}

public int GetMax(int[] array) { //ищет максимальное число в массиве

int max = array[0];

for (int i = 1; i < array.Length; ++i) {

if (array[i] > max)

max = array[i];

}

return max;

}

public static void factorial(int n) {

if (n < 0)

Console.WriteLine("Некоректно введённые данные");

else {

int f = 1;

for (int i = 2; i <= n; ++i) {

f = f \* i;

}

Console.WriteLine($"Факториал = {f}");

}

}

public void simple(int n) {

int fals = 0;

for (int i = 2; i <= n/2; ++i) {

Console.WriteLine(n % i);

if (n % i == 0) {

fals++;

Console.WriteLine("Число не простое");

break;

}

}

if (fals == 0)

Console.WriteLine("Число простое");

}

}

class Greeting {

string str;

public Greeting() {

str = "Hello";

}

public Greeting(string str)

{

this.str = str;

}

public void print() {

Console.WriteLine(str);

}

}

class Program {

static void Main() {

int n = Convert.ToInt32(Console.ReadLine());

int[] numbers = new int[n];

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

numbers[i] = Convert.ToInt32(Console.ReadLine());

}

int min = Function.GetMin(numbers);

Console.WriteLine($"Минимальное число = {min}");

Function max = new Function();

Console.WriteLine($"Максимальное число = {max.GetMax(numbers)}");

int fact = Convert.ToInt32(Console.ReadLine());

Function.factorial(fact);

int num = Convert.ToInt32(Console.ReadLine());

Function obj = new Function();

obj.simple(num);

Greeting g,g1, g2, g3;

g = new Greeting();

g1 = new Greeting("Good morning");

g2 = new Greeting("Good evening");

g3 = new Greeting("Good night");

g.print();

g1.print();

g2.print();

g3.print();

}

}

}

**Вывод с консоли**

5

3677

37

-35

772

8342

Минимальное число = -35

Максимальное число = 8342

13

Факториал = 1932053504

37

1

1

1

2

1

2

5

1

7

4

1

11

9

7

5

3

1

Число простое

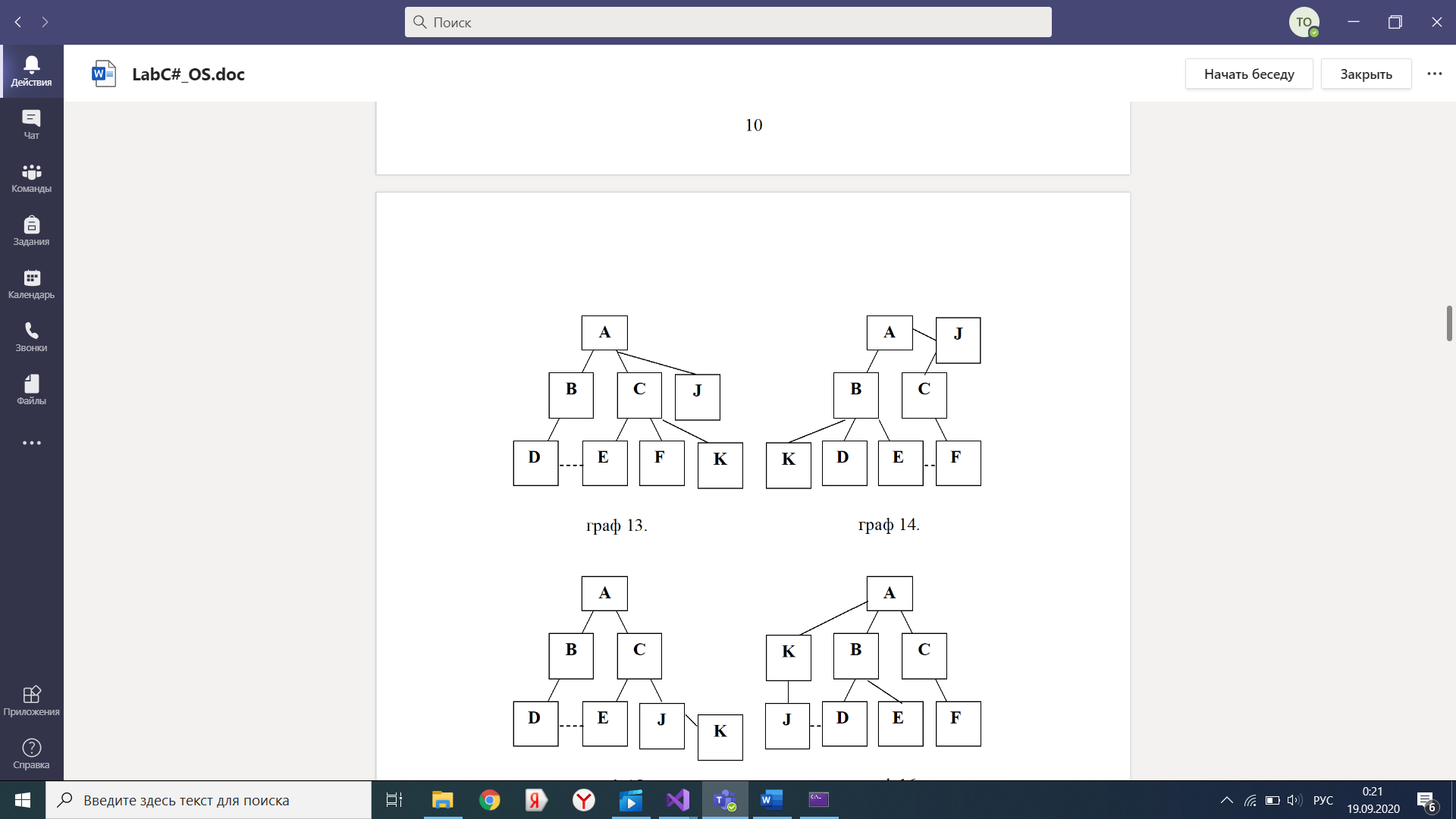
Hello

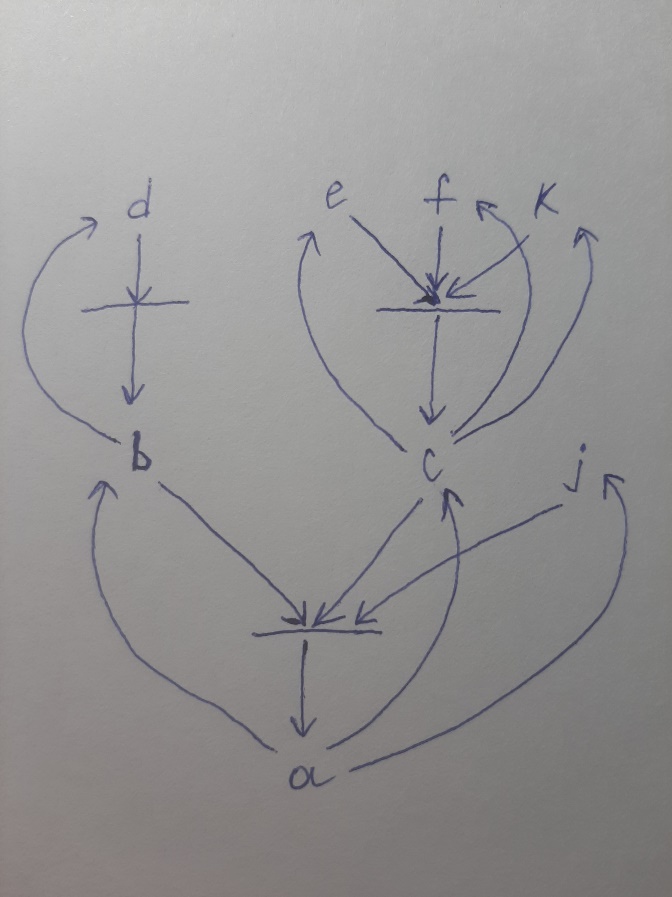
Good morning

Good evening

Good night

**Граф**





**Программа 2 (пункт 4)**

using System;

namespace Lab1\_2 {

class Program {

class A {

B objB = null;

public B b {

get {

Console.Write("a -> b ");

return objB;

}

}

C objC = null;

public C c {

get {

Console.Write("a -> c ");

return objC;

}

}

J objJ = null;

public J j {

get {

Console.Write("a -> j ");

return objJ;

}

}

public A(B b, C c, J j) {

objB = b;

objC = c;

objJ = j;

}

public void methA() {

Console.WriteLine("hash={0} Method A", this.GetHashCode());

}

}

class B {

D objD = null;

public D d {

get {

Console.Write("-> d ");

return objD;

}

}

public B(D d) {

objD = d;

}

public void methB() {

Console.WriteLine("hash={0} Method B", this.GetHashCode());

}

}

class C {

E objE = null;

public E e {

get {

Console.Write("-> e ");

return objE;

}

}

F objF = null;

public F f {

get {

Console.Write("-> f ");

return objF;

}

}

K objK = null;

public K k {

get {

Console.Write("-> k ");

return objK;

}

}

public C(E e, F f, K k) {

objE = e;

objF = f;

objK = k;

}

public void methC() {

Console.WriteLine("hash={0} Method C", this.GetHashCode());

}

}

class J {

public J() { }

public int vJ = 0;

public J(int val) {

vJ = val;

}

public void methJ() {

Console.WriteLine("hash={0} Method J", this.GetHashCode());

}

}

class D {

public D() { }

public int vD = 7;

public D(int val) {

vD = val;

}

public void methD() {

Console.WriteLine("hash={0} Method D", this.GetHashCode());

}

}

class E {

public E() { }

public int vE = 202;

public E(int val) {

vE = val;

}

public void methE() {

Console.WriteLine("hash={0} Method E", this.GetHashCode());

}

}

class F {

public F() { }

public int vF = 67;

public F(int val) {

vF = val;

}

public void methF() {

Console.WriteLine("hash={0} Method F", this.GetHashCode());

}

}

class K {

public K() { }

public int vK = 31;

public K(int val) {

vK = val;

}

public void methK() {

Console.WriteLine("hash={0} Method K", this.GetHashCode());

}

}

static void Main() {

K objK = new K();

F objF = new F();

E objE = new E();

D objD = new D();

J objJ = new J();

C objC = new C(objE, objF, objK);

B objB = new B(objD);

A objA = new A(objB, objC, objJ);

objA.methA();

objB.methB();

objA.b.methB();

objC.methC();

objA.c.methC();

objJ.methJ();

objA.j.methJ();

objD.methD();

objA.b.d.methD();

objE.methE();

objA.c.e.methE();

objF.methF();

objA.c.f.methF();

objK.methK();

objA.c.k.methK();

}

}

}

**Вывод с консоли**

hash=58225482 Method A

hash=54267293 Method B

a -> b hash=54267293 Method B

hash=18643596 Method C

a -> c hash=18643596 Method C

hash=33574638 Method J

a -> j hash=33574638 Method J

hash=33736294 Method D

a -> b -> d hash=33736294 Method D

hash=35191196 Method E

a -> c -> e hash=35191196 Method E

hash=48285313 Method F

a -> c -> f hash=48285313 Method F

hash=31914638 Method K

a -> c -> k hash=31914638 Method K

**Вывод**

Я научилась создавать возвратные и невозвратные функции, статические и нестатические. Также научилась делать агрегацию по ссылке. Объект “а” является целым, а все остальные объекты его части в соответствии с иерархией. Так например мы связываем объект а с объектами b, c, j с помощью конструктора, где b, c, j являются параметрами.