**Урок практического обучения**

**Обратное преобразование алгоритма**

**Инструменты: VisualStudio, он-лайн компилятор, графический редактор DrawIO**

[**https://www.onlinegdb.com/online\_csharp\_compiler**](https://www.onlinegdb.com/online_csharp_compiler)

**Задание 1. Дано выражение. Составьте блок-схему и программный код для его вычисления.**

|  |
| --- |
| **4 вариант** |
|  |

using System;

class Hello

{

static void Main()

{

Console.WriteLine("введите число");

int a = int.Parse(Console.ReadLine());

double b = 0;

if (a < 1)

{

b = Math.Log(Math.Abs(5.3 \* Math.Pow(a, 3 \* a) - Math.Pow(a, 2)));

Console.WriteLine(b);

}

else if (a>=2 && a<=4)

{

b = 0.025 / (Math.Tan(2.6 + a));

Console.WriteLine(b);

}

else if(a>5)

{

b=Math.Pow(Math.Sin(a - 6), 2);

Console.WriteLine(b);

}else

{

Console.WriteLine(b);

}

}

}

**Задание 2. Поменяйтесь рабочими местами с другой группой разработчиков. Исследуйте программный код. Определите наличие отклонений от алгоритма. Для исследования кода используйте метод ревьюирования и обратного преобразования кода. Составьте отчет о ревьюировании кода в форме:**

**Тестировщик: Ф.И.**

**Дата обзора:**

**Найденные ошибки:**

|  |  |
| --- | --- |
| **Номер строки** | **Описание ошибки** |
|  |  |

**Рекомендации по внедрению кода:**

**Задание 3. Дано описание объектов предметной области. Составьте диаграмму классов для реализации кода с учетом того, что объекты связаны отношением наследования. Объекты отличаются друг от друга хотя бы одним полем. Объекты имеют как минимум два поля. Напишите ООП-код в консольном исполнении. При написании кода учтите, что пользователь с использованием созданного вами меню должен выбрать объект и метод с которым он будет выполнять операции.**

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Классы** | **Методы** |
|  | Студент, преподаватель, персона, заведующий кафедрой | **Ввод(); Печать()** |
|  | Организация, страховая компания, нефтегазовая компания, завод |
|  | Игрушка, продукт, товар, медикаменты |
|  | Товар, велосипед, квадроцикл, самокат |
|  | Тест, экзамен, выпускной экзамен, испытание |
|  | Животное, домашнее, дикое, курица, медведь |
|  | Музыкант, персона, студент, гитарист |
|  | Грузовик, автомобиль, легковое авто, транспорт |
|  | Окружность, геометрическая фигура, линия, заливка |
|  | Квитанция, накладная, документ, счет |

using System;

using System.Security.Cryptography;

class G

{

public int sum;

public string inf;

public void vv()

{

Console.WriteLine("сумма товара");

int x = int.Parse(Console.ReadLine());

Console.WriteLine("о товаре");

string c = Console.ReadLine();

sum = x; inf = c;

}

public void vv2()

{

Console.WriteLine("сумма = " + sum + "\nИнформация о товаре = "+inf);

}

}

class В:G

{

int f;

public void vv()

{

Console.WriteLine("сумма товара");

int x = int.Parse(Console.ReadLine());

Console.WriteLine("о товаре");

string c = Console.ReadLine();

Console.WriteLine("код");

int c1=int.Parse(Console.ReadLine());

sum = x; inf = c; f = c1;

}

public void vv2()

{

Console.WriteLine("сумма = " + sum + "\nИнформация о товаре = " + inf);

}

}

class К : G

{

int bat;

public void vv()

{

Console.WriteLine("сумма товара");

int x = int.Parse(Console.ReadLine());

Console.WriteLine("о товаре");

string c = Console.ReadLine();

Console.WriteLine("код");

int c1 = int.Parse(Console.ReadLine());

sum = x; inf = c; bat = c1;

}

public void vv2()

{

Console.WriteLine("сумма = " + sum + "\nИнформация о товаре = " + inf);

}

}

class С : G

{

int name;

public void vv()

{

Console.WriteLine("сумма товара");

int x = int.Parse(Console.ReadLine());

Console.WriteLine("о товаре");

string c = Console.ReadLine();

Console.WriteLine("код");

int c1 = int.Parse(Console.ReadLine());

sum = x; inf = c; name = c1;

}

public void vv2()

{

Console.WriteLine("сумма = " + sum + "\nИнформация о товаре = " + inf);

}

}

class f

{

static void Main()

{

G p = new G();

В p1 = new В();

К p2 = new К();

char wq;

do

{

Console.WriteLine("что вы хотите?");

Console.WriteLine("0-товар\n1-велосипед\n2-квадроцикл\n3-самокат");

int x = int.Parse(Console.ReadLine());

switch (x)

{

case 0:

Console.WriteLine("что вы хотите сделать?");

Console.WriteLine("0-ввод\n1-вывод");

int g = int.Parse(Console.ReadLine());

switch (g)

{

case 0: p.vv(); break;

case 1: p.vv2(); break;

}

break;

case 1:

Console.WriteLine("что вы хотите сделать?");

Console.WriteLine("0-ввод\n1-вывод");

int g1 = int.Parse(Console.ReadLine());

switch (g1)

{

case 0: p1.vv(); break;

case 1: p1.vv2(); break;

}

break;

case 2:

Console.WriteLine("что вы хотите сделать?");

Console.WriteLine("0-ввод\n1-вывод");

int g2 = int.Parse(Console.ReadLine());

switch (g2)

{

case 0: p2.vv(); break;

case 1: p2.vv2(); break;

}

break;

case 3:

С p3 = new С();

Console.WriteLine("что вы хотите сделать?");

Console.WriteLine("0-ввод\n1-вывод");

int g3 = int.Parse(Console.ReadLine());

switch (g3)

{

case 0: p3.vv(); break;

case 1: p3.vv2(); break;

}

break;

}

Console.WriteLine("закончился ввод?");

Console.WriteLine("нет-n\nда-y");

wq = char.Parse(Console.ReadLine());

} while (wq == 'y');

}

}

**Задание 4. Поменяйтесь рабочими местами с другой группой разработчиков. Исследуйте программный код. Определите наличие отклонений от алгоритма. Для исследования кода используйте метод ревьюирования и обратного преобразования кода. Составьте отчет о ревьюировании кода в форме:**

**Тестировщик: Ф.И.**

**Дата обзора:**

**Найденные ошибки:**

|  |  |
| --- | --- |
| **Номер строки** | **Описание ошибки** |
|  |  |