

ТЕМА 1 ПРОСТЫЕ ОПЕРАТОРЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ C#

1.1 Цель первой темы

Знакомство с основами среды Visual Studio.NET и приобретение практических навыков по созданию консольных приложений. Освоение программирования на языке C# с использованием простых операторов.

1.2 Теоретические сведения

1.2.1 Введение

Язык C# является наиболее известной новинкой в области языков программирования. Это первый язык программирования, созданный в 21-м веке. Язык признан международным сообществом. В июне 2006 года Европейской ассоциацией по стандартизации принята уже четвертая версия стандарта этого языка: Standard ECMA-334 C# Language Specifications, 4-th edition.

Руководителем группы, создающей язык C#, является сотрудник Microsoft Андреас Хейлсберг. Он был известен в мире программистов задолго до того, как пришел в Microsoft. Хейлсберг входил в число ведущих разработчиков одной из самых популярных визуальных сред программирования Delphi.

Можно отметить следующие основные достоинства языка C#:

- язык C# создавался и развивается параллельно с каркасом Framework.Net и в полной мере учитывает все его возможности;
- язык C# является полностью объектно-ориентированным языком;
- благодаря каркасу Framework.Net, ставшему надстройкой над операционной системой, программисты языка C# получают преимущества работы с виртуальной машиной;
- Framework.Net поддерживает разнообразие типов приложений на языке C#;
- реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху языка C#.

1.2.2 Система типов языка C#

Стандарт языка C# включает следующий набор фундаментальных типов:

- логический тип (bool);
- символьный тип (char);
- целые типы. Целые типы могут быть одного из трех размеров - short, int, long, сопровождаемые описателем signed или unsigned, который указывает, как интерпретируется значение - со знаком или без него;
- вещественные типы. Эти типы также могут быть одного из трех размеров - float, double, longdouble;
- тип void, используемый для указания на отсутствие информации.

Язык позволяет конструировать типы:

- указатели (например, `int*` - типизированный указатель на переменную типа `int`);
- ссылки (например, `double&` - типизированная ссылка на переменную типа `double`);
- массивы (например, `char[]` - массив элементов типа `char`).

Язык позволяет конструировать пользовательские типы:

- перечислимые типы (`enum`) для представления значений из конкретного множества;
- структуры (`struct`);
- классы (`class`).

Описание основных типов языка C# приведено в таблице 1.1

Таблица 1.1 Основные типы данных языка C#

Имя типа	Системный тип	Значения	Размер
<code>bool</code>	<code>System.Boolean</code>	<code>true</code> , <code>false</code>	8 бит
<code>sbyte</code>	<code>System.SByte</code>	<code>[-128, 127]</code>	Знаковое, 8-бит
<code>byte</code>	<code>System.Byte</code>	<code>[0, 255]</code>	Беззнаковое, 8-бит
<code>short</code>	<code>System.Short</code>	<code>[-32768, 32767]</code>	Знаковое, 16-бит
<code>ushort</code>	<code>System.UShort</code>	<code>[0, 65535]</code>	Беззнаковое, 16-бит
<code>int</code>	<code>System.Int32</code>	$[-2^{31}, 2^{31}]$	Знаковое, 32-бит
<code>uint</code>	<code>System.UInt32</code>	$[0, 2^{32}]$	Беззнаковое, 32-бит
<code>long</code>	<code>System.Int64</code>	$[-2^{63}, 2^{63}]$	Знаковое, 64-бит
<code>ulong</code>	<code>System.UInt64</code>	$\approx [0, 2^{64}]$	Беззнаковое, 64-бит
<code>float</code>	<code>System.Single</code>	$[10^{-45}, 10^{38}]$	7 цифр
<code>double</code>	<code>System.Double</code>	$[10^{-324}, 10^{308}]$	15-16 цифр
<code>decimal</code>	<code>System.Decimal</code>	$[10^{-28}, 10^{28}]$	28-29 значащих цифр
<code>char</code>	<code>System.Char</code>	<code>[U+0000, U+ffff]</code>	16-бит Unicode символ
<code>string</code>	<code>System.String</code>	Строка из символов Unicode	

1.2.3 Выражения

Выражения строятся из операндов – констант, переменных, функций объединенных знаками операций и скобками. При вычислении выражения определяется его значение и тип. Эти характеристики выражения однозначно определяются значениями и типами операндов, входящих в выражение, и правилами вычисления выражения. Правила задаются:

- приоритетом операций (для операций одного приоритета порядок применения – слева направо или справа налево);
- преобразование типов операндов и выбор реализации для перегруженных операций;
- тип и значение результата выполнения операции над заданными значениями операндов определенного типа.

Приоритет и порядок выполнения операций приведен в таблице 1.2

Таблица 1.2 Приоритет и порядок выполнения некоторых операций в С#

Приоритет	Категория	Операции	Порядок
0	Первичные	x.y, f(x), a[x], x++, x--, new	→
1	Унарные	+, -, !, ++x, --x, (T)x	→
2	Мультипликативные	*, /, %	→
3	Аддитивные	+, -	→
4	Сдвиг	<<, >>	→
5	Отношения, проверка типов	<, >, <=, >=, is, as	→
6	Эквивалентность	==, !=	→
7	Логическое И (AND)	&	→
8	Логическое ИЛИ (OR)		→
9	Условное логическое И	&&	→
10	Условное логическое ИЛИ		→
11	Условное выражение	? :	←
12	Присваивание	=, *=, /=, %=, +=, -=	←

Любое выражение, взятое в скобки, получает высший приоритет и должно быть вычислено, прежде чем к нему будут применимы какие-либо операции. Скобки позволяют изменить стандартный порядок вычисления выражения и установить порядок, необходимый для вычисления в данном конкретном случае. В сложных выражениях скобки полезно расставлять даже в том случае, если стандартный порядок совпадает с требуемым, поскольку наличие "лишних" скобок зачастую увеличивает наглядность записи выражения.

1.2.4 Преобразование типов

Операция кастинга - приведения операндов к одному типу. В ходе вычисления выражения может возникнуть необходимость выполнения преобразования типов операндов. По возможности эти преобразования выполняются автоматически, неявно для программиста. Но неявные преобразования ограничены, поскольку могут быть только безопасными. Когда же нужно выполнить опасное преобразование, программист должен задать его явно. Одна из возможностей явного задания преобразования типа состоит в применении операции приведения к типу, называемой также кастингом. Это унарная операция со следующим синтаксисом:

(тип) выражение;

Здесь в скобках указывается тип, к которому следует привести выражение, стоящее за скобками. Нужно понимать, что не всегда существует явное приведение типа источника к типу цели. Операция кастинга применима только для приведения типов внутри арифметического типа. С ее помощью один арифметический подтип можно привести к другому подтипу, но нельзя, например, целочисленные типы привести к логическому типу bool.

Рассмотрим примеры приведения типа:

```
byteb1 = 1, b2 = 2, b3;  
//b3 = b1 + b2;  
b3 = (byte)(b1 + b2);
```

В этом примере необходимо сложить две переменные типа `byte` и, казалось бы, никакого приведения типов выполнять не нужно, результат будет также иметь тип `byte`, согласованный с левой частью оператора присваивания. Однако это не так по той простой причине, что отсутствует операция сложения над короткими числами. Реализация сложения начинается с типа `int`. Поэтому перед выполнением сложения оба операнда неявно преобразуются к типу `int`, результат сложения будет иметь тип `int`, и при попытке присвоить значение выражения переменной типа `byte` возникнет ошибка периода компиляции. По этой причине оператор во второй строке кода закомментирован. Программист должен явно привести выражение к типу `byte`, что и демонстрирует третья строка кода, в которой использована операция приведения к типу.

В следующем фрагменте кода демонстрируется еще один пример приведения типа:

```
int i1,i2;  
i1 = -40;  
i2 = (int)(1.8 * i1) + 32;
```

Результат умножения имеет тип `double` по типу первого операнда. Перед тем как выполнять сложение, результат приводится к типу `int`. После приведения типа для первого слагаемого `(int)(1.8 * i1)` сложение будет выполняться над целыми числами, результат будет иметь тип `int`, и не потребуются никаких преобразований для присвоения полученного значения переменной `i2`. Если убрать приведение типа в этом операторе, то возникнет ошибка на этапе компиляции.

1.3 Пример выполнения задания на лабораторную работу

Написать программу, которая меняет местами значения переменных `a` и `b`. Исходное значение переменной `a` формируется случайным образом в диапазоне от 0 до 20, а значение переменной `b` вводится в режиме диалога. Предусмотреть вывод значений переменных до и после обмена их значений. Программу выполнить в консольном приложении, которое разместить на рабочем столе компьютера.

Запускаем среду программирования Visual Studio 2022(см. Рисунок 1.1).

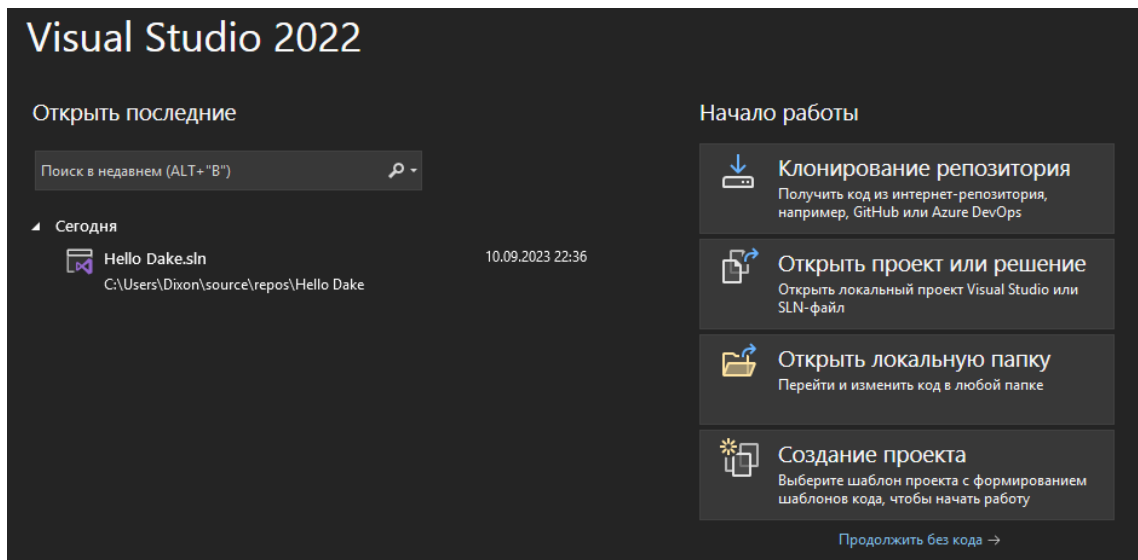


Рисунок 1.1 – Окно 1 – запуска Visual Studio.NET

В окнереисунка 1.1 выбираем команды File/New/Project...

Открывается новое окно (NewProject), в котором можно задать тип создаваемого проекта (смотри рисунок 1.2).

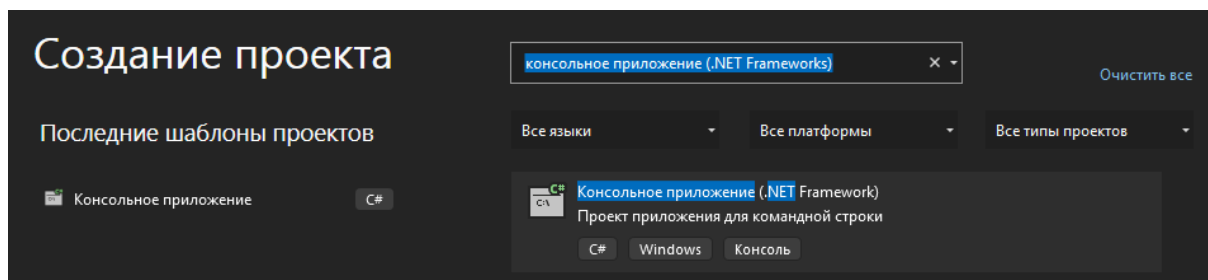


Рисунок 1.2 – Окно 2 – создания нового проекта

В окне 2 выбираем тип проекта ConsoleApplication и задаем место создания папки, в которой будут размещаться файлы создаваемого проекта (Рабочий стол папка ИДЗ_1). Остальные настройки окна можно не менять.

Нажимаем кнопку ОК и переходим к окну 3.

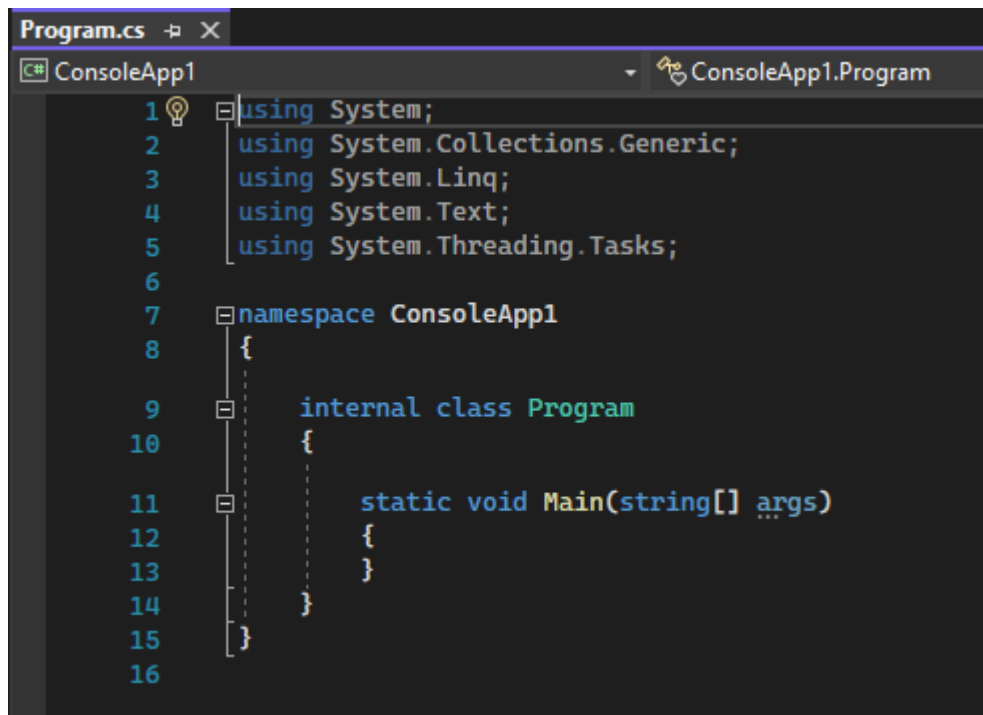


Рисунок 1.3 – Окно 3 среды Visual Studio.NET

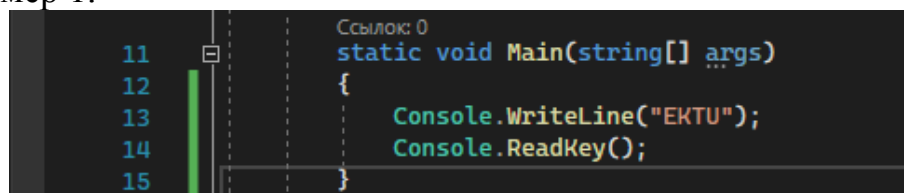
В окне 3 на странице редактора Program.cs будет набираться программный код решения задачи.

Алгоритм решения задачи очень простой:

- необходимо организовать ввод в режиме диалога значения переменной a;
- сформировать случайным образом в заданном диапазоне значение переменной b;
- напечатать исходные значения этих переменных;
- выполнить обмен значений этих переменных;
- напечатать новые значения переменных a и b.

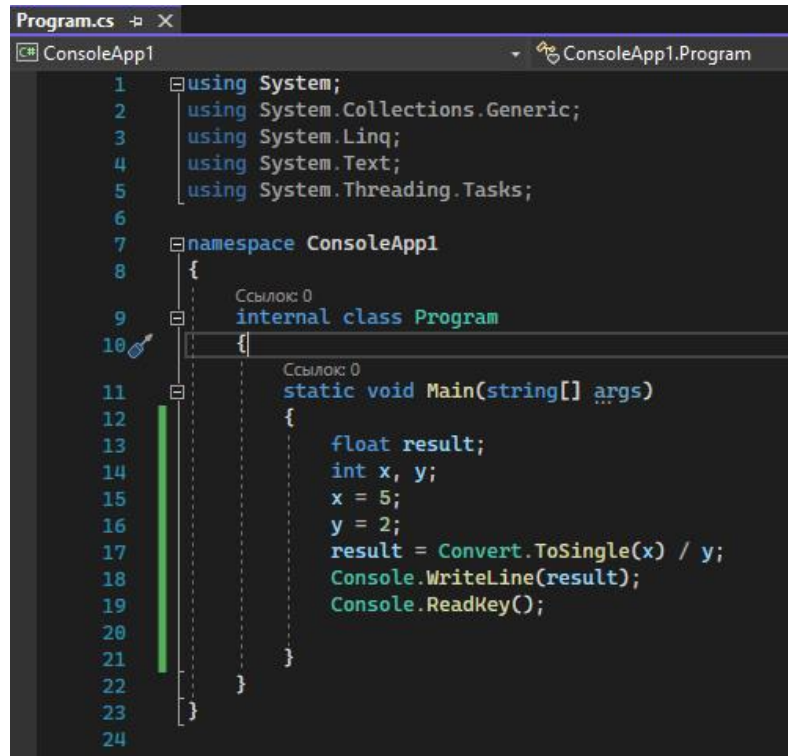
Для реализации перечисленных пунктов алгоритма необходимо изучить материал первых двух лекций дисциплины.

Пример 1.



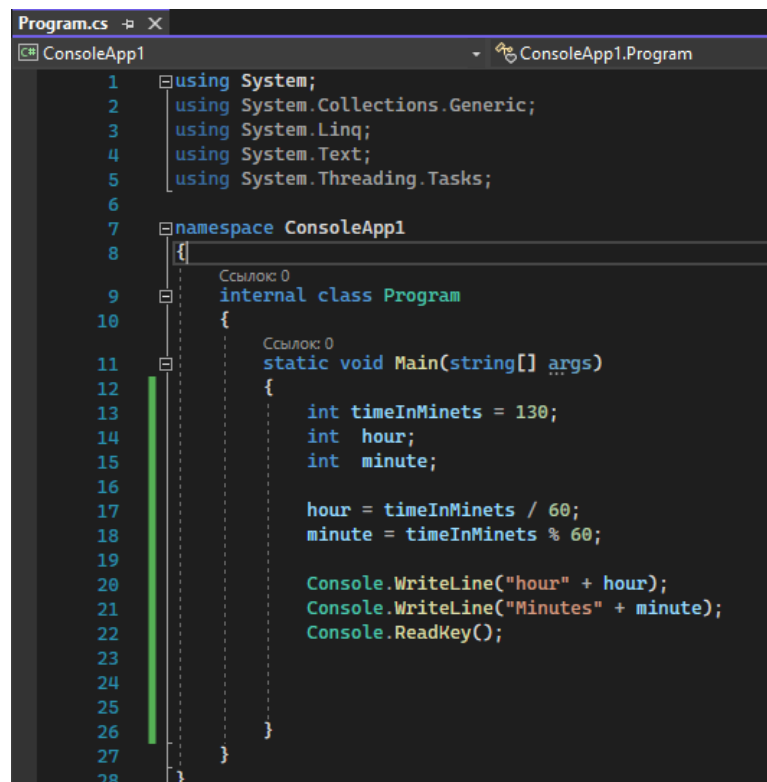
Пример 2.

Convert.ToSingle Метод. Преобразует заданное значение в число с плавающей запятой одиночной точности.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApp1
8 {
9     internal class Program
10     {
11         static void Main(string[] args)
12         {
13             float result;
14             int x, y;
15             x = 5;
16             y = 2;
17             result = Convert.ToSingle(x) / y;
18             Console.WriteLine(result);
19             Console.ReadKey();
20         }
21     }
22 }
23
24
```

Пример 3 Преобразовать количество минут (timeInMinets) в часы и минуты, а затем вывести результат на консоль.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApp1
8 {
9     internal class Program
10     {
11         static void Main(string[] args)
12         {
13             int timeInMinets = 130;
14             int hour;
15             int minute;
16
17             hour = timeInMinets / 60;
18             minute = timeInMinets % 60;
19
20             Console.WriteLine("hour" + hour);
21             Console.WriteLine("Minutes" + minute);
22             Console.ReadKey();
23         }
24     }
25 }
26
27
28
```

Самостоятельные задачи:

1 Задание по программированию: Дележ яблок-1

N школьников делят K яблок поровну, не делящийся остаток остается в корзинке. Сколько яблок достанется каждому школьнику?

1	Тест 1
2	Входные данные:
3	3
4	14
5	
6	
7	Вывод программы:
8	4

2 Задание по программированию: Дележ яблок-2

N школьников поделили K яблок поровну, не делящийся остаток остался в корзинке. Сколько яблок осталось в корзинке?

1	Тест 1
2	Входные данные:
3	3
4	14
5	
6	
7	Вывод программы:
8	2

3 Задание по программированию: Последняя цифра

Дано натуральное число. Выведите его последнюю цифру.

1	Тест 1
2	Входные данные:
3	179
4	
5	Вывод программы:
6	9

4 Задание Пусть есть два товара, первый из них стоит A тенге B тиын, а второй - C тенге D тиын. Сколько тенге и тиын стоят эти товары вместе.

В задачах где есть несколько размерностей величин (например, тенге и тиын, километры и метры, часы и минуты) следует переводить все в наименьшую единицу измерения, осуществлять необходимые действия, а затем переводить обратно к нужным единицам.

В нашей задаче наименьшей единицей являются тиын, поэтому все цены следует перевести в них, затем сложить их, а затем перевести результат обратно в теңге и тиын.

Следующим этапом индивидуального задания является построение структурной схемы алгоритма решения задачи.

5 Задание по программированию: Первая цифра двузначного числа.

Дано положительное двузначное число. Найдите число десятков в нем.

1	Тест 1
2	Входные данные:
3	42
4	
5	Вывод программы:
6	4

6 Задание по программированию: Сумма цифр трехзначного числа

Дано трехзначное число. Найдите сумму его цифр.

7 Задание по программированию: Последняя цифра

Дано натуральное число. Выведите его последнюю цифру.

1	Тест 1
2	Входные данные:
3	179
4	
5	Вывод программы:
6	9

8 Задание



Рисунок 1.4 – Структурная схема алгоритма решения задачи

Используя структурную схему алгоритма решения задачи, разрабатываем код программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            int a, b, c;
            string buf;
            //Ввод значения переменной a в режиме диалога
            Console.Write("Введите целое значение a ");
            buf = Console.ReadLine();
            a = Convert.ToInt32(buf);
            //Формирование случайным образом значения переменной b
            Random rnd = new Random();
            b = rnd.Next() % 21;
            //Печать исходных значений
            Console.WriteLine("Исходные значения переменных a и b:");
            Console.WriteLine("a={0} b={1}", a, b);
            //обмен
            c = a; a = b; b = c;
            //Печать значений переменных a и b после обмена
            Console.WriteLine("Новые значения переменных a и b:");
            Console.WriteLine("a={0} b={1}", a, b);
            // Задержка рабочего экрана монитора
            Console.WriteLine("Для продолжения нажмите клавишу Enter");
            Console.ReadLine();
        }
    }
}
```

Работа программы:

Введите целое значение а 35

Исходные значения переменных а и b:

a=35 b=13

Новые значения переменных а и b:

a=13 b=35

Для продолжения нажмите клавишу Enter

При оформлении отчета по лабораторной работе рекомендуется следующая структура и последовательность элементов:

- титульный лист;
- название лабораторной работы;
- цель лабораторной работы;
- индивидуальное задание на лабораторную работу;
- краткие комментарии по выполнению индивидуального задания (при необходимости структурная схема алгоритма решения задачи);
- необходимый программный код индивидуального задания;
- результаты работы программы;
- выводы.

Титульный лист является первой страницей отчета и служит источником информации, необходимой для поиска документа. Поэтому он содержит название министерства, название университета, название кафедры, название дисциплины, название модуля дисциплины, ФИО студента, выполнившего лабораторную работу и ФИО преподавателя, принимающего отчет по лабораторной работе. Внизу титульного листа указывается место и год выполнения лабораторной работы, например, Усть-Каменогорск 2023г.

Титульный лист включают в общую нумерацию страниц отчета, но номер страницы на титульном листе не проставляется.

Название лабораторной работы должно соответствовать названию из методических указаний по выполнению лабораторных работ, часто оно соответствует названию модуля.

Цель лабораторной работы содержит краткое описание цели соответствующего модуля дисциплины.

Индивидуальное задание на лабораторную работу содержит полный текст индивидуального задания, полученного у преподавателя только после выполнения домашнего задания по лабораторной работе.

Краткие комментарии по выполнению индивидуального задания содержат описание алгоритма выполнения индивидуального задания. При необходимости приводится структурная схема алгоритма или его подробное словесное описание.

Необходимый программный код индивидуального задания содержит либо полный текст кода программы, либо фрагменты кода программы, разработанные студентом и коды добавляемые средой программирования, без которых объяснение выполненной работы невозможно.

Результаты работы программы обычно содержат копии окон работы программы во всех ее режимах.

В выводах обычно отмечается результат выполнения лабораторной работы.

Страницы текста отчета должны соответствовать формату А4.

Печатаение отчета выполняется машинописным способом или с применением печатающих и графических устройств вывода ЭВМ на одной стороне листа белой бумаги. Тип шрифта - TimesNewRoman, основной размер шрифта - № 14, допускается № 12. Основной интервал -1, допускается -1,5.

Текст отчета следует печатать, соблюдая следующие разделы полей: правое, верхнее, нижнее и левое – 20 мм.

Абзацный отступ начинается не менее чем с четвертого знака и должен быть одинаков в пределах одного документа.

Вне зависимости от способа выполнения отчета, качество напечатанного текста, иллюстраций, таблиц и приложений должно удовлетворять требованию их чёткого воспроизведения.

Вписывать в отпечатанный текст отчета отдельные слова, формулы и знаки допускается только черными чернилами или черной тушью, при этом плотность вписанного текста должна быть максимально приближена к плотности основного текста.

Опечатки, описки и графические неточности допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного изображения машинописным способом или от руки черными чернилами или чёрной тушью.

Повреждения листов отчета, пометки и следы не полностью удалённого текста не допускаются.

Разделы должны иметь порядковые номера в пределах всего отчета, обозначенные арабскими цифрами без точки и записанные с отступом. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделённых точкой. В конце номера подраздела точки не ставятся. Разделы, как и подразделы, могут состоять из одного или нескольких пунктов.

Разделы, подразделы должны иметь заголовки. Пункты заголовков не имеют.

Заголовки разделов документа следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчёркивая и не выделяя.

Заголовки подразделов документа следует располагать в середине строки и печатать строчными буквами, начиная с первой прописной, выделяя жирным шрифтом.

Если раздел не имеет подразделов, то нумерация пунктов в нем должна быть в пределах этого раздела, и номер пункта должен состоять из номеров раздела и пункта, разделённых точкой. В конце номера пункта точка не ставится.

Если раздел имеет подразделы, то нумерация пунктов должна быть в пределах каждого подраздела и номер пункта должен состоять из номеров раздела, подраздела и пункта, разделённых точками.

Страницы отчета следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту.

Номер страницы отчета проставляют по центру страницы вверху без точки в конце.

Страницы отчета скрепляются скрепкой или размещаются в файле (степлер не использовать).

1.4 Домашнее задание на лабораторную работу

Написать программу вычисления длины отрезка, заданного координатами его концов $A(x_1, y_1)$ и $B(x_2, y_2)$. Значения координат точки A вводить в режиме диалога, а точки B задавать случайным образом в диапазоне минус 100 плюс 100. Предусмотреть вывод результатов работы программы на экран монитора.

1.5 Индивидуальные задания для СРС

1.5.1 Индивидуальное задание 1 студента:

1.5.1.1 Написать программу вычисления выражения:

$$Y=3*(X+1)^3+4*(X-1)+2.$$

Значение X задавать в режиме диалога. Методом перебора попытаться найти такое X при котором Y равен нулю. Предусмотреть вывод результата на экран монитора.

1.5.1.2 Написать программу, которая в вещественном числе, введённом в режиме диалога, выделяет целую и дробную части, и отдельно выводит их на экран монитора.

1.5.1.3 В режиме диалога заданы координаты вершин треугольника. Найти периметр треугольника.

1.5.2 Индивидуальное задание 2 студента:

1.5.2.1 Написать программу вычисления суммы:

$$S=1+1/2+1/3+1/4+1/5.$$

Предусмотреть вывод результата на экран монитора.

1.5.2.2 Написать программу вычисления площади треугольника, стороны которого A , B , C вводятся в режиме диалога. Предусмотреть вывод результата на экран монитора.

1.5.2.3 Написать программу вычисления длины ломаной линии состоящей из трех отрезков, заданных координатами их концов $A(x,y)$, $B(x,y)$, $C(x,y)$ и $D(x,y)$. Значения координат формировать случайным образом в диапазоне от 0 до 100.

1.5.3 Индивидуальное задание 3 студента:

1.5.3.1 Написать программу вычисления выражения:

$$C = A^2 + \sqrt{(A + \ln B) / 2}.$$

Значения переменных А и В, необходимые для решения данного выражения, вводятся в режиме диалога.

- 1.5.3.2 Написать программу, которая реализует алгоритм обмена значениями переменных А и В (без использования промежуточной переменной с). Исходные значения переменных вводить в режиме диалога. Предусмотреть вывод результата до и после обмена.
- 1.5.3.3 Написать программу вычисления площади круга. Значение радиуса вводится в режиме диалога с ЭВМ и должно быть больше нуля.
- 1.5.4 Индивидуальное задание 4 студента:
 - 1.5.4.1 Длина отрезка задана в дюймах (1 дюйм = 2,54 см.), вводится в режиме диалога. Перевести значение длины в метрическую систему, то есть выразить её в метрах, сантиметрах и миллиметрах. Так, например, 21 дюйм = 0м 53см 3,4мм.
 - 1.5.4.2 Длина окружности задается в режиме диалога. Найти площадь круга, ограниченного этой окружностью.
 - 1.5.4.3 В режиме диалога заданы координаты двух противоположных вершин прямоугольника. Найти площадь прямоугольника.
- 1.5.5 Индивидуальное задание 5 студента:
 - 1.5.5.1 Значение Х задается в режиме диалога. Используя только арифметические операции вычислить выражение

$$Y = 1 - 2 * X + 3 * X^2 - 4 * X^3.$$
 Разрешается использовать не более восьми операций в одном выражении.
 - 1.5.5.2 Угол α задаётся в режиме диалога в градусах, минутах и секундах. Найти его величину в радианах.
 - 1.5.5.3 Длина ребра куба задается в режиме диалога. Найти объем куба и площадь его поверхностей.
- 1.5.6 Индивидуальное задание 6 студента:
 - 1.5.6.1 Найти площадь кольца, внешний радиус которого равен 100, а внутренний задается в режиме диалога и должен быть меньше 100.
 - 1.5.6.2 Длина стороны равностороннего треугольника задается в режиме диалога. Найти площадь этого треугольника.
 - 1.5.6.3 Смешано v_1 литров воды температуры t_1 с v_2 литрами воды температуры t_2 . Найти объем и температуру образовавшейся смеси. Значения v_1 , t_1 , v_2 и t_2 задаются в режиме диалога.

1.6 Контрольные вопросы для защиты отчета на СРСП

- 1.6.1 Что входит в понятие платформы.NET?
- 1.6.2 Что означает понятие общезыковая среда выполнения (Common Language Runtime, CLR)?
- 1.6.3 Что понимается под термином «приложение»?
- 1.6.4 Что понимается под термином «проект»? Его состав?

- 1.6.5 Что включает в себя «Пространство имен»?
- 1.6.6 Что включает в себя среда Visual Studio.NET ?
- 1.6.7 Понятие алгоритма? Основные свойства алгоритма.
- 1.6.8 Этапы решения задачи?
- 1.6.9 Какие типы данных языка C# Вы знаете?
- 1.6.10 Какие знаки операций языка C# Вы знаете?
- 1.6.11 Формат записи оператора присваивания языка C#? Пояснить его работа на примере.
- 1.6.12 Как можно организовать ввод данных «в режиме диалога»? Пример.
- 1.6.13 Как можно организовать вывод данных на экран монитора? Пример.
- 1.6.14 Как можно организовать вывод данных на экран монитора в заданном формате? Пример.
- 1.6.15 Как формируется случайное число в заданном диапазоне? Пример.