Лабораторная работа №2

Использование основных операторов языка C#

1. Цель задания:

1) Получение практических навыков использования операторов выбора.
2) Получение практических навыков выбора и использования операторов циклов.

2. Теоретические сведения

Операторы управления работой программы называют управляющими конструкциями программы. К ним относят:

- оператор-выражение;
- составные операторы;
- операторы выбора;
- операторы циклов;
- операторы перехода.

2.1. Оператор – выражение

Любое выражение, завершающееся точкой с запятой, рассматривается как оператор, выполнение которого заключается в вычислении выражения. Частным случаем выражения является пустой оператор.

```
i++; //инкремент
x+=y+x;//аддитивное присваивание
t=a>b;// присваивание результата отношения
; //пустой оператор
```

2.2. Составной оператор

Составной оператор (блок) – это последовательность описаний и операторов, заключенная в фигурные скобки. Блок воспринимается компилятором как один оператор.

```
{
n++;
summa+=n;
}
```

2.2. Операторы выбора

Операторы выбора – это условный оператор и переключатель.

1. Условный оператор имеет полную и сокращенную форму.

```
if (выражение-условие) оператор;   //сокращенная форма
```

В качестве выражения-условия могут использоваться арифметическое выражение, отношение и логическое выражение. Если значение выражения-условия отлично от нуля (т. е. истинно), то выполняется оператор.

```
if (x<y&&x<z)min=x;
```

```
if (выражение-условие) оператор1;  //полная форма
else оператор2;
```

Если значение выражения-условия отлично от нуля, то выполняется оператор1, при нулевом значении выражения-условия выполняется оператор2.

```
if (d>=0)
{
x1=(-b-sqrt(d))/(2*a);
x2=(-b+sqrt(d))/(2*a);
Console.WriteLine(”x1={0},x2={1}”,x1,x2);
}
else cout<<“\nРешения нет”;
```

2. Переключатель определяет множественный выбор.

```
switch (выражение)
{
case константа1 : оператор1 ;
case константа2 : оператор2 ;
. . . . . .
[default: операторы;]
}
```

При выполнении оператора `switch`, вычисляется выражение, записанное после `switch`, оно должно быть целочисленным. Полученное значение последовательно сравнивается с константами, которые записаны следом за `case`. При первом же совпадении выполняются операторы, помеченные данной меткой. Если выполненные операторы не содержат оператора перехода, то далее выполняются операторы всех следующих вариантов, пока не появится оператор перехода или не закончится переключатель. Если значение выражения, записанного после `switch`, не совпало ни с одной константой, то выполняются операторы, которые следуют за меткой `default`. Метка `default` может отсутствовать.

```
int i;
string buf;
Console.Write(”Введите число:”);
buf=Console.ReadLine();
i=int.Parse(buf);
switch(i)
{
     case 1: Console.WriteLine("\nthe number is one");break;
     case 2: Console.WriteLine ("\n2*2="+i*i);break;
     case 3: Console.WriteLine("\n3*3="+i*i);break;
     case 4: Console.WriteLine ("\n"+i+" is very beautiful!");
             break;

     default: WriteLine ("\nThe end of work");
}
```

Результаты работы программы:

1. При вводе 1 будет выведено:

```
The number is one
```

2. При вводе 2 будет выведено:

```
2*2=4
```

3. При вводе 3 будет выведено:

```
3*3=9
```

4. При вводе 4 будет выведено:

```
4 is very beautiful!
```

5. При вводе всех остальных чисел будет выведено:

```
The end of work
```

2.3. Операторы циклов

- Цикл с предусловием:

```
while (выражение-условие)
оператор;
```

В качестве <выражения-условия> чаще всего используется отношение или логическое выражение. Если оно истинно, т. е. не равно 0, то тело цикла выполняется до тех пор, пока выражение-условие не станет ложным.

```
Console.Write("?  ");
buf = Console.ReadLine();
a = int.Parse(buf);
while (a != 0)
     {
          if (a % 2 == 0) s += a;
          Console.Write("? ");
          buf = Console.ReadLine();
          a = int.Parse(buf);
     }
```

- Цикл с постусловием:

```
do
оператор
while (выражение-условие);
```

Тело цикла выполняется до тех пор, пока выражение-условие истинно.

```
do
{
  Console.Write("? ");
  buf = Console.ReadLine();
  a = int.Parse(buf);
  if (a % 2 == 0 && a != 0) s += a;
} while (a != 0);
```

- Цикл с параметром:

```
for (выражение_1;выражение-условие;выражение_3)
оператор;
```

выражение_1 и выражение_3 могут состоять из нескольких выражений, разделенных запятыми. Выражение_1 – задает начальные условия для цикла (инициализация). Выражение-условие определяет условие выполнения цикла, если оно не равно 0, цикл выполняется, а затем вычисляется значение выражения_3. Выражение_3 – задает изменение параметра цикла или других переменных (коррекция). Цикл продолжается до тех пор, пока выражение-условие не станет равно 0. Любое выражение может отсутствовать, но разделяющие их « ; » должны быть обязательно.

```
1.
for (int i = 0; i < n; i++)
{
     Console.WriteLine("?");
     buf = Console.ReadLine();
     a = int.Parse(buf);
     if (a % 2 == 0) s += a;
}
```

```
2. // Изменение шага корректировки
for ( n=2; n>60; n+=13)
{
     оператор;
}
```

```
3.//проверка условия отличного от того, которое налагается на
//число итераций
for ( num=1;num*num*num<216; num++)
{
     оператор;
}
```

```
4. //коррекция с помощью умножения
for ( d=100.0; d<150.0;d*=1.1)
{
     оператор;
 }
```

```
5.//коррекция с помощью арифметического выражения
for (x=1;y<=75;y=5*(x++)+10)
{
     оператор;
}
```

```
6. //использование нескольких корректирующих выражений, тело
//цикла отсутствует
for (x=1, y=0; x<10;x++,y+=x);
```

Часто переменная, которая управляет циклом for, необходима только для этого цикла и больше никак не используется. В этом случае можно объявить ее в разделе инициализации цикла. Например, следующая программа вычисляет как сумму, так и факториала чисел от 1 до 5. Управляющая переменная i здесь объявляется в цикле for.

```
public static void Main()
{
int sum = 0;
int fact = 1;
// Вычисляем сумму и факториал чисел от 1 до 5.
```

```
for(int i = 1; i <= 5; i++)
{
     sum += i; // i известна только в пределах цикла.
     fact *= i;
}
// Но здесь переменная i неизвестна.
Console.WriteLine("Сумма равна " + sum);
Console.WriteLine("Факториал равен " + fact);
}
```

2.4. Операторы перехода

Операторы перехода выполняют безусловную передачу управления. В C# есть пять операторов, изменяющих естественный порядок выполнения вычислений:

- оператор безусловного перехода `goto;`
- оператор выхода из цикла `break;`
- оператор перехода к следующей итерации цикла `continue;`
- оператор возврата из функции `return ;`
- оператор генерации исключения `throw`.

- `break` – оператор прерывания цикла.

```
{
     оператор;
     if (<выражение_условие>) break;
     оператор;
}
```

Т. е. оператор `break` целесообразно использовать, когда условие продолжения итераций надо проверять в середине цикла.

```
// Найти сумму чисел, числа вводятся с клавиатуры до тех пор,
пока не будет //введено 100 чисел или 0.
for(s=0, i=1; i<100;i++)
{
     Console.WriteLine("?");
     buf = Console.ReadLine();
     x = int.Parse(buf);
     // если ввели 0, то суммирование заканчивается
     if( x==0) break;
     s+=x;
}
```

- `continue` – переход к следующей итерации цикла. Он используется, когда тело цикла содержит ветвления.

```
//Найти количество и сумму положительных чисел
for( k=0,s=0,x=1;x!=0;)
{
     Console.WriteLine("?");
     buf = Console.ReadLine();
     x = int.Parse(buf);
     if (x<=0) continue;
     k++; s+=x;
```

```
}
```

- `goto <метка>` – передает управление оператору, который содержит метку.

  В теле той же функции должна присутствовать конструкция: `<метка>:оператор;`

Метка – это обычный идентификатор, областью видимости которого является функция. Оператор `goto` передает управления оператору, стоящему после метки. Использование оператора `goto` оправдано, если необходимо выполнить переход из нескольких вложенных циклов или переключателей вниз по тексту программы или перейти в одно место функции после выполнения различных действий.

Применение `goto` нарушает принципы структурного и модульного программирования, по которым все блоки, из которых состоит программа, должны иметь только один вход и только один выход.

Нельзя передавать управление внутрь операторов `if, switch` и циклов. Нельзя переходить внутрь блоков, содержащих инициализацию, на операторы, которые стоят после инициализации.

- `return` – оператор возврата из функции. Он всегда завершает выполнение функции и передает управление в точку ее вызова. Вид оператора:
  `return [выражение];`

3. Постановка задачи

Решить указанные в варианте задачи, используя основные операторы языка C#. При решении задачи, использовать все типы циклов (for, while, do while).

1. Дана последовательность из n целых чисел. Найти среднее арифметическое этой последовательности.
2. Дана последовательность из n целых чисел. Найти сумму четных элементов этой последовательности.
3. Дана последовательность из n целых чисел. Найти сумму элементов с четными номерами из этой последовательности.
4. Дана последовательность из n целых чисел. Найти сумму нечетных элементов этой последовательности.
5. Дана последовательность из n целых чисел. Найти сумму элементов с нечетными номерами из этой последовательности.
6. Дана последовательность из n целых чисел. Найти минимальный элемент в этой последовательности.
7. Дана последовательность из n целых чисел. Найти номер максимального элемента в этой последовательности.
8. Дана последовательность из n целых чисел. Найти номер минимального элемента в этой последовательности.
9. Дана последовательность из n целых чисел. Найти максимальный элемент в этой последовательности.
10. Дана последовательность из n целых чисел. Найти сумму минимального и максимального элементов в этой последовательности.
11. Дана последовательность из n целых чисел. Найти разность минимального и максимального элементов в этой последовательности.
12. Дана последовательность из n целых чисел. Найти количество нечетных элементов этой последовательности.
13. Дана последовательность из n целых чисел. Найти количество четных элементов этой последовательности.

14. Дана последовательность из n целых чисел. Найти количество элементов этой последовательности, кратных числу К.
15. Дана последовательность из n целых чисел. Найти количество элементов этой последовательности, кратных ее первому элементу.
16. Дана последовательность из n целых чисел. Найти количество элементов этой последовательности, кратных числу K1 и не кратных числу K2.
17. Дана последовательность из n целых чисел. Определить, каких чисел в этой последовательности больше: положительных или отрицательных.
18. Дана последовательность целых чисел, за которой следует 0. Найти среднее арифметическое этой последовательности.
19. Дана последовательность целых чисел, за которой следует 0. Найти сумму четных элементов этой последовательности.
20. Дана последовательность целых чисел, за которой следует 0. Найти сумму элементов с четными номерами из этой последовательности.
21. Дана последовательность целых чисел, за которой следует 0. Найти сумму нечетных элементов этой последовательности.
22. Дана последовательность целых чисел, за которой следует 0. Найти сумму элементов с нечетными номерами из этой последовательности.
23. Дана последовательность целых чисел, за которой следует 0. Найти минимальный элемент в этой последовательности.
24. Дана последовательность целых чисел, за которой следует 0. Найти номер максимального элемента в этой последовательности.
25. Дана последовательность целых чисел, за которой следует 0. Найти номер минимального элемента в этой последовательности.
26. Дана последовательность целых чисел, за которой следует 0. Найти максимальный элемент в этой последовательности.
27. Дана последовательность целых чисел, за которой следует 0. Найти сумму минимального и максимального элементов в этой последовательности.
28. Дана последовательность целых чисел, за которой следует 0. Найти разность минимального и максимального элементов в этой последовательности.
29. Дана последовательность целых чисел, за которой следует 0. Найти количество нечетных элементов этой последовательности.
30. Дана последовательность целых чисел, за которой следует 0. Найти количество четных элементов этой последовательности.
31. Дана последовательность целых чисел, за которой следует 0. Найти количество элементов этой последовательности, кратных числу К.
32. Дана последовательность целых чисел, за которой следует 0. Найти количество элементов этой последовательности, кратных ее первому элементу.
33. Дана последовательность целых чисел, за которой следует 0. Найти количество элементов этой последовательности, кратных числу K1 и не кратных числу K2.
34. Дана последовательность целых чисел, за которой следует 0. Определить, каких чисел в этой последовательности больше: положительных или отрицательных.
35. $S = 1 - 2 + 3 - 4 + 5 - \ldots$, всего n слагаемых;
36. S=1+3+5+7+ …, всего n слагаемых;
37. S=1+2-3+4+5-6+7+8-9+…, всего n слагаемых;
38. S=15+17-19+21+23-25+…, всего n слагаемых;
39. $S = \sin X + \sin X^2 + \sin X^3 + \sin X^4 + \ldots + \sin X^n$
40. $S = \sin X + \sin^2 X + \sin^3 X + \sin^4 X + \ldots + \sin^n X$
41. $S = \sqrt{3 + \sqrt{6 + \sqrt{9 + \ldots \sqrt{99}}}}$
42. $S = \sin(x + \cos(2x - \sin(3x + \cos(4x + \sin(5x - \cos(6x + \ldots)\ldots)$

43. Найти первое отрицательное число последовательности u=cos(ctg(n)), где n=1,2,3….
44. Определить является ли число k степенью 3.
45. Определить является ли число k простым.
46. Дана последовательность из 100 чисел. Найти номер первого отрицательного числа.
47. Найти количество цифр в десятичном числе k.
48. Найти сумму цифр в десятичном числе k.
49. Сформировать n чисел Фибоначчи ($a_1=1$, $a_2=1$,$a_i=a_{i-1}+a_{i-2}$).
50. Сформировать все числа Фибоначчи не превышающие заданное число Q.
51. Дано число k. Определить, является ли оно числом Фибоначчи.
52. $P = \frac{2}{3} \cdot \frac{4}{5} \cdot \frac{6}{7} \cdot \ldots \cdot \frac{2N}{2N+1}$.
53. $P = a \cdot (a+1) \cdot \ldots \cdot (a+n-1)$.
54. $S = \frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \ldots \frac{1}{a^{2n-1}}$.
55. $P = \frac{(x-1)(x-3)(x-7)\ldots\ldots(x-63)}{(x-2)(x-4)(x-8)\ldots\ldots(x-64)}$.
56. $P = (1+\sin 0{,}1)(1+\sin 0{,}2)\ldots\ldots(1+\sin 10)$.
57. $P = (1-\frac{1}{2^2})(1-\frac{1}{3^2}) \cdot \ldots \cdot (1-\frac{1}{n^2})$, где n>2.
58. $P = (1-\frac{1}{2})(1-\frac{1}{4})(1-\frac{1}{6}) \cdot \ldots \cdot (1-\frac{1}{2n})$
59. $S = \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \ldots + \frac{1}{(2n+1)^2}$

4. Варианты

Вариант	Задача 1	Задача 2	Задача 3
1	1	34	35
2	2	33	36
3	3	32	37
4	4	31	38
5	5	30	39
6	6	29	40
7	7	28	41
8	8	27	42
9	9	26	43
10	10	25	44
11	11	24	45
12	12	23	46
13	13	22	47
14	14	21	48
15	15	20	49
16	16	19	50
17	17	18	51
18	1	23	52
19	2	24	53
20	3	25	54
21	4	26	55

22	5	27	56
23	6	28	57
24	7	29	58
25	8	30	59

5. Методические указания

1. Ввод данных в задачах №1и №2 осуществляется с клавиатуры.
2. Массивы при решении задач не используются.
3. При решении задачи №1 целесообразно использовать цикл с параметром, т. к. известно количество элементов последовательности.
4. При решении задачи №2 целесообразно использовать цикл с условием, т. к. известно, что признаком окончания последовательности является 0.

6. Содержание отчета

1. Постановка задач для конкретного варианта.
2. Алгоритм решения каждой задачи в виде блок-схемы.
3. Программы для решения задач на языке C#.
4. Тесты (с проверкой достаточности по критериям черного ящика и МГТ) для каждой задачи.