

Работа 1.1. Алгоритмическая конструкция “ветвление”

Цель: формирование навыков реализации алгоритмов ветвления

Введение

Алгоритмическая конструкция **ветвление** используется для выбора одной из двух или нескольких альтернатив. Имеет два основных варианта.

Развилка реализуется с помощью условного оператора **if**:

```
if (условие)
    { блок_1 }
[ else { блок_2 } ]
```

Сначала проверяется условие. Если оно верно, выполняется блок_1, иначе выполняется блок_2.

В условиях используют операции сравнения, которые в языке C# записываются так: == равно, != не равно, < , > , <=, >=; а также логические операции && “и”, || “или”, ! отрицание “не”.

Блоки могут содержать фрагменты кода, заключенные в фигурные скобки, которые можно опустить, если фрагмент состоит из одного выражения.

Переключатель реализуется с помощью оператор выбора **switch**:

```
switch (выражение)
{
    case выражение_1: блок_1 ; break;
    case выражение_2: блок_2 ; break;
    .....
    case выражение_n: блок_n ; break;
    [ default: блоки_nnn; ]
}
```

Пример 1. Использование конструкции **if ... else**

для проверки знака и четности введенного числа

- Создадим проект **con111**. Наберем код программы в теле метода **Main**:

```
string mes1 = "", mes2 = "";           // объявление и инициализация
Console.WriteLine("Введите целое число ");
// ввод строки и преобразование в целое число
int n = int.Parse(Console.ReadLine());
if (n>=0) mes1 = "положительное";
else mes1 = "отрицательное";
if (n%2==0) mes2 = "четное";
else mes2 = "нечетное";
Console.WriteLine("Число {0} {1} {2}", n, mes1, mes2);
Console.ReadKey();
```

- Протестируем программу (**F5**). Откорректируем программный код.

В случаях выбора и присваивания только одного значения из двух альтернативных вариантов вместо оператора `if` удобно использовать тернарный оператор условного присваивания (`= ? :`)

`= (условие) ? вариант1 : вариант2 ;`

При отладке программ часто используют случайные числа. Методы их генерации описаны в классе **System.Random**. Для получения псевдослучайной последовательности чисел сначала создается экземпляр класса **Random**, затем методом **Next(n1,n2)** генерируется число в диапазоне [n1,n2], например:

```
Random rnd = new Random();    int n = rnd.Next(-20, 70);
```

Пример 2. Использование тернарного оператора
для проверки знака и четности случайного числа

- Создадим проект **con112**. Наберем код программы в теле метода **Main**:

```
Random rnd = new Random();           // создаем экземпляр класса Random
// генерируем случайное целое число в интервале -40, 40
int n = rnd.Next(-40, 40);
// используем тернарный оператор
string mes1 = (n >= 0) ? "положительное" : "отрицательное";
string mes2 = (n % 2 == 0) ? "четное" : "нечетное";
Console.WriteLine("Число {0} {1} {2}", n, mes1, mes2);
Console.ReadKey();
```
- Протестируем программу. Откорректируем программный код.
При необходимости выбора из нескольких вариантов удобно использовать конструкцию **switch ...case**

Пример 3. Использование конструкции выбора `switch...case`

- Создадим проект **con113**. Наберем код программы в теле метода **Main**:

```
string mes;
Console.WriteLine("Введите день недели: Пн, Вт, Ср, Чт, Пт, Сб, Вс");
string day = Console.ReadLine();
switch (day)
{
    case "Сб": mes = "Иду в гости"; break;
    case "Вс": mes = "Отдыхаю"; break;
    default:  mes = "Работаю"; break;
}
Console.WriteLine(mes);
Console.ReadKey();
```
- Протестируем программу. Откорректируем программный код.

Пример 4*. Проверка логина и пароля, введенных с клавиатуры.

Пусть эталонные логин длиной не меньше 6 символов и пароль хранятся в строковых переменных **myLog** и **myPas** в коде программы. Сначала проверяем длину и совпадение введенного логина с заданным. В случае правильности логина проверяем совпадение пароля.

- Создадим проект **con114**. Наберем код программы в теле метода **Main**:

```
// задаем эталонные логин и пароль, а также выводимые сообщения
string myLog = "qwerty", myPas = "asdf";
string mes = "", mesYes = "Добро пожаловать!", mesNo = "Вход воспрещен!";
Console.Write("Введите логин: ");
string log = Console.ReadLine(); // ВВОДИМ ЛОГИН
// проверяем длину и совпадение логина
if (log.Length < 6) mes = "Логин очень короткий!\n" + mesNo;
else if (log != myLog) mes = "Неверный логин!\n" + mesNo;
else
// в случае истинности логина вводим и проверяем пароль
{ Console.Write("Введите пароль: ");
  string pas = Console.ReadLine(); // ВВОДИМ ПАРОЛЬ
  mes = (pas == myPas) ? mesYes : "Неверный пароль!\n" + mesNo;
}
Console.WriteLine(mes); // ВЫВОДИМ СООБЩЕНИЯ
Console.ReadKey();
```

- Протестируем программу. Откорректируем программный код.

Пример 5*. Простейший калькулятор на 4 действия

- Создадим проект **con115**. Наберем код программы в теле метода **Main**:

```
// инициализируем переменные res и ok.
double A, B, res=0; bool ok=true;
Console.Write("Введите число A: ");
string sA = Console.ReadLine(); // ВВОД СТРОКИ
A = double.Parse(sA); // ПРЕОБРАЗОВАНИЕ В ЧИСЛО
Console.Write("Введите число B: ");
string sB = Console.ReadLine();
B=double.Parse(sB);
Console.Write("Введите знак операции (+-*/) ");
string op = Console.ReadLine();

switch (op)
{ case "+": res = A + B; break;
  case "-": res = A - B; break;
  case "*": res = A * B; break;
  case "/": res = A / B; break;
  default: ok = false; break;
}
if (ok)
  Console.WriteLine("{0} {1} {2} = {3}", A, op, B, res); //ВЫВОД
else Console.WriteLine("Недопустимая операция");
Console.ReadKey();
```

- Протестируем программу. Откорректируем программный код.

В рассмотренном примере простейшего калькулятора 5* полезно производить проверку вводимых данных и, если введено не число, присваивать значение по умолчанию, например 1.

Фрагмент кода с проверкой и преобразованием строки в число примет вид:

```
if (!double.TryParse(sA, out A))
    { Console.WriteLine("Не число!"); A = 1; }
if (!double.TryParse(sB, out B))
    { Console.WriteLine("Не число!"); B = 1; }
```

- Протестируем окончательный вариант.

✂ Задания для самостоятельной работы

Создайте консольные приложения, в которых выполняются заданные действия:

1. Генерируется случайное целое число **n** в интервале [20 – 99].
Проверяется, делится ли оно на введенное с клавиатуры целое число **d** (используйте тернарный оператор).
2. Генерируются два случайных целых числа **a** и **b** в интервале [10 – 30].
Если **a > b**, выводится их разность, если **a < b**, выводится сумма, если **a == b**, выводится сообщение “числа равны”. (оператор **if**)
3. Генерируются два случайных целых числа **c** и **d** в интервале [-10 – 10].
Если оба числа четные, выводится их частное, если оба нечетные - их сумма, если одно число четное, а другое нечетное, выводится их произведение (**if**),
4. По введенному номеру месяца выводится название поры года (зима, весна, лето, осень) и сообщение: сессия, каникулы, 1 семестр, 2 семестр (**if**).
5. Проверяется соответствие веса и роста (ввод рост и вес; вывод одного из сообщений "Норма", "Нужно похудеть", "Нужно поправиться".
Нормальный вес (в кг) = (рост (в см) – 100) ± 10%. (оператор **if**)
6. Введенная цифра (от 0 до 5) выводится прописью (оператор **switch**).
7. Выводится название предмета по введенной первой букве: ф - физика, м - математика, и - история, г - география, б - биология (оператор **switch**).
8. Выводится название страны и ее столицы по введенной первой букве:
б – Беларусь, Минск, р – Россия, Москва, л –Литва, Вильнюс (**switch**),
9. По введенному номеру выводится название дня недели (1 – пн, 2 –вт,...) и сообщение “рабочий день” или “выходной” (операторы **switch**, **if**).
10. Проверяется правильность введенного логина строго из 5 букв и пароля из 6 и более цифр.