

Семинар 4. Управляющие операторы, часть 2

4.1 Операторы ветвления

4.1.1 Конструкция $x = \text{условие} ? x_1 : x_2$

При написании различных выражений программисты довольно часто сталкиваются с ситуациями, когда значение некоторой величины зависит от определенного условия. Например, если нам нужно взять наибольшее из двух чисел, мы можем написать

```
Int32 a=5, b=7, max;
if (a > b)
    max = a;
else
    max = b;
```

В «сишных» языках специально для этой конструкции есть сокращенная форма записи, которая не только короче, но и со временем кажется значительно понятнее. Выражение начинается с условия, которое заключается в круглые скобки. После условия ставится знак вопроса и перечисляются два варианта, первый – на случай, если условие верно, второй – если оно ложно. Варианты отделяются друг от друга символом двоеточия.

```
Int32 a=5, b=7, max;
max = (a > b) ? a : b;
```

Какой вариант ты будешь использовать – зависит исключительно от тебя самого. Но знать, о том, что есть сокращенный синтаксис, и уметь его читать просто необходимо. Иначе ты не сможешь легко общаться с другими программистами, которые этой формой записи пользуются.

4.1.2 Конструкция *switch*

Зачастую при составлении условий возникает необходимость расширить стандартный набор ответов «да» и «нет». Чтобы отреагировать на ответы «не знаю» или «мне все равно», можно конечно использовать серию условных операторов

```
if (answer == "да")
{
    // действия для ответивших да
}
else if (answer == "нет")
{
    // действия для ответивших нет
}
else if (answer == "незнаю")
{
    // действия для незнаек
}
else if (answer == "мне все равно")
{
    // действия для тех кому все равно
}
else
{
    // действия для всех остальных
}
```

Но получается это неуклюже, поэтому лучше воспользоваться специальной конструкцией для выбора из множества взаимоисключающих условий. Она называется `switch` и работает следующим образом:

```
switch(answer)
{
    case "да":
        // действия для ответивших да
        break;
    case "нет":
        // действия для ответивших нет
        break;
    case "незнаю":
        // действия для незнаек
        break;
    case "мне пофигу":
        // действия для пофигистов
        break;
    default:
        // действия, когда переменная имеет значение,
        // отличающееся от всех вышеперечисленных
}
```

В скобках после служебного слова «`switch`» записана переменная, которая будет использоваться далее для сравнений. Каждое сравнение начинается со слова «`case`», после которого нужно написать условие и поставить двоеточие. В конце каждого блока нужно напечатать команду «`break`», которая выполнит преждевременный выход из конструкции.

Обычно оператор `break` недолюбливают за то, что он ломает логику работы программы, и потом трудно мысленно восстановить алгоритм, но это один из немногих случаев, когда его использование является обязательным и оправдано на 100%.

Следует заметить, что оператор `switch` является лишь только сокращенной формой записи рассмотренной ранее конструкции. По факту компилятор языка создаст один и тот же код, поэтому никакой прибавки в скорости это не даст.

4.2 Имитация искусственного интеллекта

Давай немного развлечемся и попробуем составить программу, размышляющую, почти как человек. Ну, или, по крайней мере, создающую такое впечатление.

Скорее всего, вы когда-нибудь встречались с интерактивными программами, как бы общающимися с людьми. Пользователь обращается за помощью к программе, а с помощью наводящих вопросов выдает готовое решение.

Для примера возьмем наверняка известную из Интернета шутливую методику решения всех проблем. Вот ее сильно сокращенный вариант:

- Вопрос 1: Ты это трогал?
«да» → следующий вопрос
«нет» → ответ программы «вот и не волнуйся» и выход
- Вопрос 2: Кто-нибудь знает?
«да» → следующий вопрос
«нет» → ответ программы «тогда нечего волноваться» и выход
- Вопрос 3: Можно свалить на другого?
«да» → ответ программы «тогда нечего волноваться» и выход
«нет» → ответ программы «ты попал =)» и выход

Метод, конечно, не блещет особой интеллектуальностью, но для начала сойдет! Осталось разобраться, как это программируется?

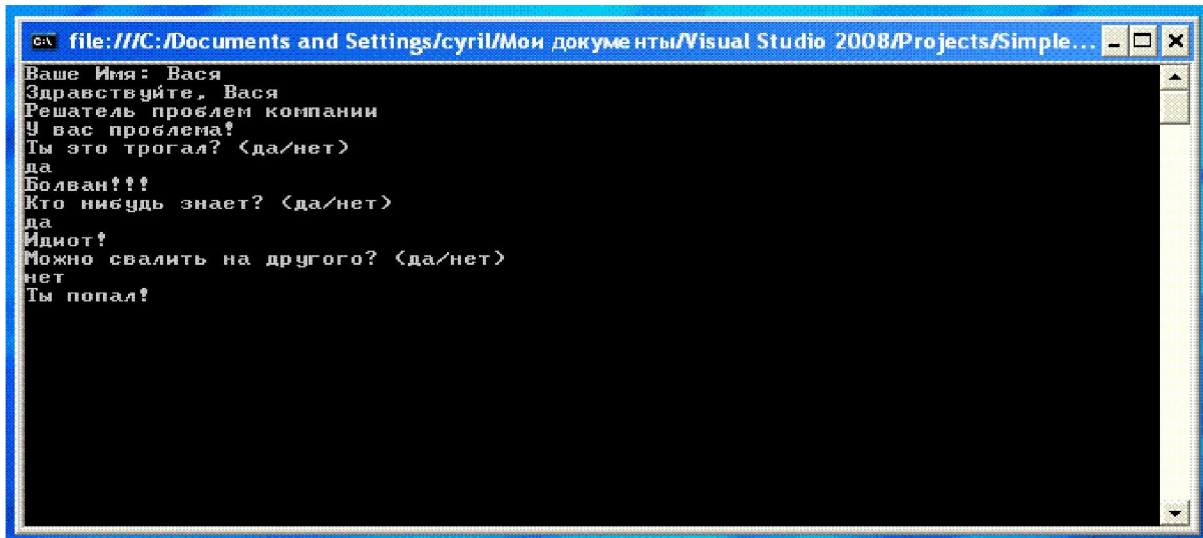


Рис. 4-1 Вывод программы «Решение проблем»

В данном случае нам намного проще писать консольную программу. Сначала нам необходимо решить вопрос взаимодействия с пользователем. Мы будем печатать вопрос на экране с помощью функции «Console.WriteLine()», далее предлагать пользователю ввести его с клавиатуры с помощью функции «Console.ReadLine()». С целью упрощения будем считать, что пользователь всегда вводит ответ маленькими русскими буквами в формате «да/нет».

```
String answer;
Console.WriteLine("Хм.. у тебя проблема...");
Console.WriteLine("Ты это трогал? да/нет");
answer = Console.ReadLine();
if (answer == "да")
{
    Console.WriteLine("Болван! ");
    Console.WriteLine("Кто-нибудь знает? да/нет");
    answer = Console.ReadLine();
    if (answer == "да")
    {
        Console.WriteLine("Идиот!!!");
        Console.WriteLine("Можно свалить на другого? да/нет");
        answer = Console.ReadLine();
        if (answer == "да")
        {
            Console.WriteLine("Вот и нет проблемы");
        }
        else
        {
            Console.WriteLine("Ты попал!!!");
        }
    }
    else
    {
        Console.WriteLine("Тогда забудь об этом");
    }
}
else
{
    Console.WriteLine("Вот и не парься");
}
Console.ReadLine();
```



Задания

#1 Создай искусственный интеллект, который сможет поддерживать разговор о компьютере пользователя:

- Компьютер: Привет, меня зовут Ксеон, а тебя как?
- Пользователь: Миша
- К: А у тебя дома есть компьютер? (да/нет)
- П: да
- К: Здорово. А какая у него производительность в МГц?
- П: 2500
- К: Хорош. У меня, правда, больше, во мне 4 ядра по 2ГГц каждое.

#2 Создай искусственный интеллект, который сможет поддерживать разговор о возрасте пользователя.

- К: Сколько вам лет?
- П: -20
- К: Вас еще нет на свете!!! Так сколько вам лет?
- П: 1 (все значения в диапазоне 1-3)
- К: Вы только родились и не можете говорить. Так сколько вам?
- П: 7 (все значения в диапазоне 4-17)
- К: Хватит играть, иди делать уроки?
- П: 18
- К: С совершеннолетием!
- П: 120 (Значения в диапазоне больше 100)
- К: Столько не живут!

#3 Создай искусственный интеллект, который сможет поддерживать разговор о домашнем любимце пользователя.

#4 Создай искусственный интеллект, который сможет поддерживать разговор о том, как прошел день в школе.

#5 Создай искусственный интеллект, который сможет поддерживать разговор о здоровье пользователя, при необходимости попросить померить температуру и сделать заключение о состоянии, здоров пользователь или болен (здоров при $36 < t < 37$)?

#6 Сделать тест, который спрашивает рост и вес и проверяет соответствие этих параметров друг другу по формуле: Вес (в кг) = Рост (в см) – 100. Если вес больше, нужно сказать, что требуется сесть на диету. Если меньше, то налегать на плюшки.

#7 Составь программу, которая запрашивает ввод трех значений температуры и проверяет, есть ли среди них температура таяния льда? (0 градусов)

- #8** Составь программу, которая запрашивает ввод трех значений ускорения и проверяет, есть ли среди них ускорение свободного падения? (9.8 м/с^2)
- #9** Составь программу, которая запрашивает ввод трех формул химических кислот и проверяет, есть ли среди них серная кислота? (H_2SO_4)
- #10** Сделай калькулятор, который делит два введенных числа друг на друга с проверкой, что второе из введенных чисел не равно нулю, и при необходимости выводит сообщение о том, что на ноль делить нельзя.