# Дисциплина «Программирование корпоративных систем» Рабочая тетрадь 1.2

**Условные конструкции языка С#**

|  |
| --- |
| **Теоретический материал** |
| **КОНСТРУКЦИЯ if..else**  Условные конструкции - один из базовых компонентов многих языков программирования, которые направляют работу программы по одному из путей в зависимости от определенных условий. Одной из таких конструкций в языке программирования C# является конструкция **if..else**  Конструкция **if..else** проверяет истинность некоторого условия и в зависимости от результатов проверки выполняет определенный код.  Ее простейшая форма состоит из блока **if**:  if (условие)  {  выполняемые инструкции  }  После ключевого слова **if** ставится условие. Условие должно представлять значение типа **bool**. Это может быть непосредственно значение типа **bool** или результат условного выражения или другого выражения, которое возвращает значение типа **bool**. И если это условие истинно (равно **true**), то срабатывает код, который помещен далее после условия внутри фигурных скобок.  Если блок **if** содержит одну инструкцию, то мы можно его сократить, убрав фигурные скобки.  Также можно соединить сразу несколько условий, используя логические операторы. |

Блок **else** выполняется, если условие после **if** ложно, то есть равно **false**. Если блок **else** содержит только одну инструкцию, то опять же можно его сократить, убрав фигурные скобки. Язык С# поддерживает синтаксис конструкции **else if**.

# КОНСТРУКЦИЯ switch

Конструкция **switch** имеет следующее формальное определение:

switch (выражение)

{

case значение1:

код,выполняемый если выражение имеет значение1 break;

case значение2:

код,выполняемый если выражение имеет значение2 break;

//.............

case значениеN:

код, выполняемый если выражение имеет значениеN break;

default:

код, выполняемый если выражение не имеет ни одно из выше указанных значений

break;

}

После ключевого слова **switch** в скобках идет сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после оператора **сase**. И если совпадение будет найдено, то будет

выполняться определенный блок **сase**. В конце каждого блока **сase** должен ставиться один из операторов перехода**: break, goto case, return** или **throw**. Как правило, используется оператор **break**. При его применении другие блоки **case** выполняться не будут.

Если необходимо, чтобы, после выполнения текущего блока **case** выполнялся другой блок **case**, то можно использовать вместо **break** оператор **goto case**:

int number = 1; switch (number)

{

case 1:

Console.WriteLine("case 1"); goto case 5; // переход к case 5

case 3:

Console.WriteLine("case 3"); break;

case 5:

Console.WriteLine("case 5"); break;

default:

Console.WriteLine("default"); break;

}

# ТЕРНАРНАЯ ОПЕРАЦИЯ

Тернарную операция позволяет проверить некоторое условие и в зависимости от его истинности выполнить некоторые действия. Она имеет

|  |  |
| --- | --- |
| следующий синтаксис:  [первый операнд - условие] ? [второй операнд] : [третий операнд]  Здесь сразу три операнда. В зависимости от условия тернарная операция возвращает второй или третий операнд: если условие равно **true**, то возвращается второй операнд; если условие равно **false**, то третий. Например:  int x=3; int y=2;  int z = x < y ? (x+y) : (x-y); Console.WriteLine(z); // 1  Здесь первый операнд (то есть условие) представляет выражение x < y.  Если оно равно **true**, то возвращается второй операнд - (x+y), если условие равно **false**, то возвращается третий операнд - (x-y).  Результат тернарной операции (то есть второй или третий операнд в зависимости от условия) присваивается переменной z. | |
|  | |
| **Задание 1** | |
| ***Задача:*** | |
|  | Реализуйте программу-календарь для месяца мая. При вводе номера дня месяца должна проводиться проверка, является ли этот день выходным или рабочим. Также добавьте возможность начать месяц с любого дня недели. Месяц может начаться с любого дня недели (для этого нужно ввести число — номер дня недели, с которого начинается месяц. Убедитесь, что введённое число корректно.).  Выходными считаются все субботы и воскресенья, а также даты  с 1 по 5 и с 8 по 10 мая включительно. |

|  |  |
| --- | --- |
|  | Пример работы:    Совет: используйте реальный календарь для проверки. |
| ***Решение:*** | |
|  | Console.Write("Введите номер дня недели, с которого начинается месяц (1-пн, ... 7-вс): ");  int dayOfThe\_Week = Convert.ToInt32(Console.ReadLine());  Console.Write("Введите день месяца: ");  int dayOfThe\_Month = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("-----Проверяем выходной ли день-----");  if ((dayOfThe\_Month >= 1 && dayOfThe\_Month <= 5) || (dayOfThe\_Month >= 8 && dayOfThe\_Month <= 10))  {      Console.WriteLine("Выходной день");  }  else  {      if ((((dayOfThe\_Month % 7) + (dayOfThe\_Week - 1)) % 7) > 5)      {          Console.WriteLine("Выходной день");      }      else      {          Console.WriteLine("Не выходной день");      }  } |
| ***Ответ:*** | |
|  |  |
| **Задание 2** | |
| ***Задача:*** | |
|  | Реализуйте программу для банкомата.  Пользователь хочет обналичить N рублей в банкомате. В банкомате присутствуют купюры достоинством 100, 200, 500, 1000, 2000 и 5000 рублей. Банкомат должен быть настроен таким образом, чтобы выдавать наименьшим количеством купюр, запрошенную сумму.  После запроса суммы программа должна выдать ответ сколько купюр каждого номинала нужно выдать (4 по 200, 1 по 1 000 и т.д.) для получения суммы ровно N рублей, либо выводить сообщение, что выдать ровно N рублей невозможно, например, если N не делится на 100.  Учтите, что банкомат не может выдать за один раз более 150 000 рублей. |
| ***Решение:*** | |
|  | Console.Write("Введите сумму для снятия: ");  int sum = 0;  bool sumIsGood = false;  while (!sumIsGood)  {      try      {          sum = Convert.ToInt32(Console.ReadLine());          if (sum % 100 != 0)          {              throw new Exception("Введена не кратная сумма, попробуйте еще раз:");          }          if (sum > 150000)          {              throw new Exception("Введена сумма больше 150.000 рублей");          }          sumIsGood = true;      }      catch (FormatException)      {          Console.WriteLine("Введенные данные некорректны, попробуйте еще раз:");      }      catch (Exception ex)      {          Console.WriteLine(ex.Message);      }  }  int banknote5000 = 0, banknote2000 = 0, banknote1000 = 0, banknote500 = 0, banknote200 = 0, banknote100 = 0;  while (sum > 0)  {      banknote5000 = sum / 5000;      sum = sum - 5000 \* banknote5000;      banknote2000 = sum / 2000;      sum = sum - 2000 \* banknote2000;      banknote1000 = sum / 1000;      sum = sum - 1000 \* banknote1000;      banknote500 = sum / 500;      sum = sum - 500 \* banknote500;      banknote200 = sum / 200;      sum = sum - 200 \* banknote200;      banknote100 = sum / 100;      sum = sum - 100 \* banknote100;  }  if (banknote5000 > 0) {      Console.WriteLine($"Купюр по 5000: {banknote5000} штук");  }  if (banknote2000 > 0) {      Console.WriteLine($"Купюр по 2000: {banknote2000} штук");  }  if (banknote1000 > 0) {      Console.WriteLine($"Купюр по 1000: {banknote1000} штук");  }  if (banknote500 > 0) {      Console.WriteLine($"Купюр по 500: {banknote500} штук");  }  if (banknote200 > 0) {      Console.WriteLine($"Купюр по 200: {banknote200} штук");  }  if (banknote100 > 0) {      Console.WriteLine($"Купюр по 100: {banknote100} штук");  } |
| ***Ответ:*** | |

|  |  |
| --- | --- |
|  |  |