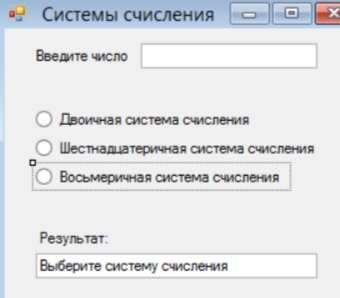
Создадим программу, которая поможет нам перевести число из десятичной системы счисления в другие самые распространённые системы: двоичную, шестнадцатеричную и восьмеричную.

Приступаем к созданию формы программы. У нас будет вот такая:



Для создания такой формы нам понадобились: 2 Label’a, 3 Radio Button’a и 2 TextBox’a. Логика программы будет построена так: пользователь вводит число в первый TextBox, нажимает на один из Radio Button’ов, чтобы выбрать систему счисления, которая будет подразумеваться под каждым из них, а во втором, нижнем, TextBox’е программа выдаст ему результат. Причём нам не понадобится нажимать на кнопки и т.п., программа будет переводить число в систему счислению по ходу того, как оно будет вписываться пользователем в первый TextBox.

Итак теперь приступим к коду. Дважды щёлкаем мышью на RadioButton1, и в том участке кода, куда нас перенесёт, пишем следующее:



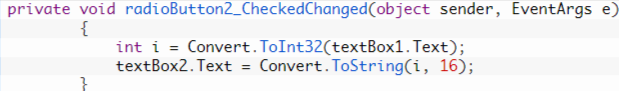
Здесь мы считываем и конвертируем в числовой целочисленный формат строку в TextBox’е, которую нам напишет пользователь. Делаем это из-за того, что, естественно, переводить в системы счисления можно только числа, но никак не строки.

Перевод в различные системы счисления в C# на Windows Forms происходит практически автоматически и в одну строчку:



Сначала разберем правую сторону. Мы конвертируем наше число обратно в строку, но, так сказать, «с оговорками» в скобках. В них i — это, понятное дело, наше число, а цифра 2 означает, что мы конвертируем значение нашей переменной i в двоичную систему счисления. Всё очень просто. Результат такой конвертации мы помещаем в TextBox2.

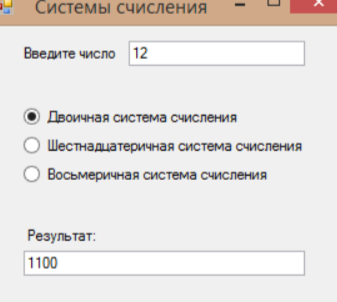
Теперь в форме щёлкаем на вторую Radio Button и пишем то же самое, но уже для шестнадцатеричной системы счисления:



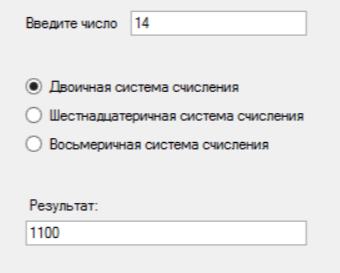
Приходится заново каждый раз объявлять переменную i из-за того, что эти переменные относятся к разным элементам управления и между собой никак не связаны и не пересекаются.

Ну и, наконец, восьмеричная система счисления на третьей Radio Button.

По сути, наша программа уже прекрасно функционирует. Мы вводим число в первый TextBox, выбираем нужную нам систему счисления — какой-либо из RadioButton’ов, а во втором TextBox’е у нас появляется результат. Мы можем щёлкать по RadioButton’ам, и ответы будут меняться в зависимости от выбранной системы счисления. Однако имеется и некоторое неудобство: если мы введём какое-нибудь другое число, которое тоже захотим перевести в другую систему счисления, то, чтобы программа заработала, надо сначала кликнуть на какой-либо RadioButton. Иначе говоря:

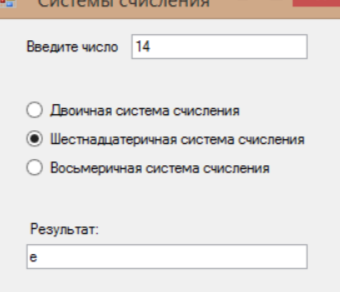


Только запустили нашу программу, ввели число «12», нажали на первый RadioButton, в результате у нас отразилось число в двоичной системе: «1100». Отлично! А теперь мы решили ещё перевести в эту же систему счисления число «14». Мы считаем «12», пишем «14», можем даже ещё раз щёлкнуть на нажатый уже RadioButton, но форма будет выглядеть вот так:

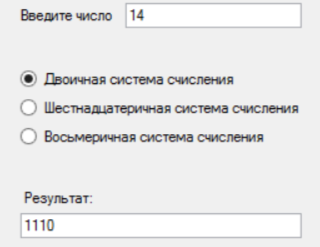


Ничего в форме, кроме вписанного нами числа, не изменилось, а результат остался от перевода числа «12».

Связано это с тем, что конвертация чисел происходит в RadioButton’ах, и происходит она при перещёлкивании кнопки между ними. Мы щёлакем на RadioButton’е, в нём срабатывает код и выводится результат. Причём, программа не принимает простого щелчка по уже нажатому RadioButton’у. Нужно обязательно нажать на другой, а потом вернуться к нему. В случае нашего примера получится примерно-так:

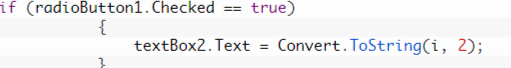


Сначала нам придётся узнать результат перевода нашего числа в шестнадцатеричную систему. И пусть там это не нужно, у нас нет другого выхода(вообще-то есть — узнать ненужный нам результат восьмеричного перевода).И вот, наконец, мы добились, чего хотели:

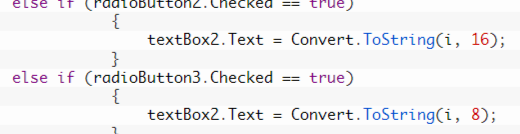


Согласитесь, не очень удобно. Поэтому исправим данное неудобство — сделаем так, что программа будет реагировать на ввод первой же цифры и сразу выдавать нам результат.

Сделаем это за счёт моментального считывания текста из TextBox1. Дважды щёлкаем на него мышью и снова объявляем переменную i. Это поможет нам считывать её моментально, без лишних щёлканий по RadioButton’ам. После этого мы создадим следующее условие:



Условие таково: если первый RadioButton имеет значение true, иначе говоря, если выбран он, то произойдёт уже известное нам действие — во втором TextBox’е у нас появится результат конвертации числа в двоичную систему. Нам осталось сделать такие же условия для оставшихся RadioButton’ов.



Если ни одно из значений не будет выбрано, то в строке результата будет написано «Выберите систему счисления» — этот текст мы написали изначально в свойствах второго TextBox’а:

