Основы программирования на C++

Урок 6. Введение в циклы: while, do while, break

Циклы while, do while

**Задача 1. Начальник.**  
  
Напишите программу для робота-начальника. Он спрашивает у пользователя, выполнил ли он задания, которые выдавал вчера, и продолжает это делать до тех пор, пока тот не ответит ему “Да, конечно, сделал”.  
  
Подсказка 1: строки можно сравнивать вот так: (answer == “Да, конечно, сделал”), где answer — переменная с типом std::string  
Подсказка 2: чтобы считать строку с пробелами, воспользуйтесь функцией std::getline(std::cin, line);"  
  
**Задача 2. Купи слона.**  
  
Напишите программу-дразнилку “Купи слона”. Она спрашивает у пользователя, как его зовут, затем отвечает “%username%, купи слона!”. Дальше, что бы он ни говорил, она передразнивает: Все говорят “...”, а ты купи слона!  
Так повторяется бесконечно.  
  
**Пример**  
  
- Хорошо, давай куплю.  
- Все говорят “Хорошо, давай куплю.”, а ты купи слона!  
  
  
**Задача 3. Таблица степеней.**  
  
Напишите программу, которая выводила бы N первых степеней числа К. Посмотрите, что случится, если N или K будут достаточно большими.  
  
**Задача 4. Напоминалка.**  
  
Василий — очень ответственный менеджер, но очень забывчивый. Иногда он пьет воду несколько раз подряд, потому что забыл, что только что пил. Напишите для него программу-напоминалку. В самом начале программа спрашивает, сколько раз ему напомнить, а затем нужное количество раз выводит “Вы хотели не забыть о чем-то”.  
  
Усложнение: добавьте возможность ввести фразу-напоминалку, чтобы можно было напоминать о чем-то конкретном.  
  
**Задача 5. Банкомат 3.0.**  
  
В банке ПлюсПлюсБанк тестируют новую систему пин-кодов на банкоматы. Теперь вместо фиксированного пин-кода пользователь должен ввести 4 числа. Если их сумма оказывается равна 42, пользователь может продолжать работу. Напишите программу, которая спрашивает у пользователя 4 числа и отвечает “Ввод корректный”, если их сумма равна 42, и “Ввод некорректный, попробуйте еще раз” в противном случае. Так повторяется бесконечно.  
  
**Задача 6. Сумма чисел.**  
  
У бухгалтера сломались счеты. Напишите программу, которая считала бы сумму вводимых чисел. В начале у бухгалтера спрашивают, сколько чисел тот хочет сложить, затем нужное количество раз просят ввести число и в конце выводят получившуюся сумму.

Применение цикла while. Инструкция break

**Задача 1. Банкомат 4.0.**  
  
В банке ПлюсПлюсБанк решили улучшить новую систему пин-кодов на банкоматы. Пользователям оказалось неудобно вводить 4 отдельных числа. Перепишите программу так, чтобы пользователь вводил число любой длины, а дальше все то же самое: если сумма цифр числа оказывается равна 42, пользователь может продолжать работу.  
  
Напишите программу, которая спрашивает у пользователя число и отвечает “Ввод корректный”, если сумма его цифр равна 42, и “Ввод некорректный” в противном случае. При этом у пользователя есть только одна попытка ввести корректное число.  
  
**Задача 2. Кадастровые проблемы.**  
  
В столице программистов на квадратные участки дается скидка. А так как работать с целыми числами программисты начинают раньше, то и участки, площадь которых является точным квадратом (например, 25 или 49), являются самыми востребованными. Напишите программу, которая проверяет, является ли введенная площадь участка точным квадратом. Самый большой участок в городе имеет площадь в 1 000 000 м2.  
  
**Задача 3. Кожаный мешок.**  
  
На этот раз нужно написать фрагмент кода для прошивки робота. Как мы знаем, роботы понимают только нули и единицы, других цифр для них не существует. Напишите программу, которая проверяет, состоит ли число только из нулей и единиц.  
  
**Задача 4. Военное время.**  
  
Во времена Второй Мировой в штабах использовались стеклянные доски. С одной их стороны специальные люди писали сводки с фронта, а с другой генералы их читали. Как вы понимаете, для генералов текст выглядел зеркально отраженным, поэтому специальных людей учили писать задом наперед. Напишите программу, упрощающую им жизнь: она должна выводить введенное число задом наперед.

Практическая работа

**В практической работе есть:**  
Обязательные задачи помогут проверить, как вы усвоили материал модуля. Их нужно сдать на проверку куратору.  
Дополнительные задачи подойдут тем, кто хочет ещё немного потренироваться, решая задачки по программированию. Вы также можете отправлять их на проверку и получать обратную связь куратора, хотя это необязательно.  
  
**Цели практической работы**  
• Научиться использовать циклы while, do while и оператор break.  
• Научиться основам работы с циклами.  
  
**Что входит в практическую работу**  
Обязательные задачи:  
1. Кукушка.  
2. Слишком большие числа.  
3. Коллекторы.  
4. Счастливый билетик.  
5. Числа Фибоначчи.  
6. Вклады.  
  
Дополнительные задачи:  
7. Сокращение дроби.  
8. Угадай число\*.

Задание 1. Кукушка

Что нужно сделать  
Напишите программу, которая имитировала бы часы с кукушкой. В начале работы она спрашивает, который час, а затем нужное количество раз пишет «Ку-ку!». При выполнении задания необходимо обеспечить контроль ввода (например, количество часов не может быть отрицательным).  
**Пример выполнения**  
  
Введите, который час: 2  
Ку-ку!  
Ку-ку!  
  
**Рекомендации по выполнению**  
Обратите внимание, если вы захотите использовать 24-часовой формат, в котором часы имеют значения 0–23, то кукушка не сможет вас оповестить о полночи (нулевой час).  
  
**Что оценивается**  
• Есть проверка входных значений.  
• Количество выведенных на консоль фраз равно количеству часов.

Задание 2. Слишком большие числа

**Что нужно сделать**  
У неудачливого бухгалтера всё опять идёт наперекосяк: ему приносят такие большие счета, что числа не помещаются на бумаге. Напишите программу, которая считала бы для него, сколько цифр (знаков) во вводимом числе.  
  
**Пример выполнения**  
  
Введите число: 10  
Цифр в числе: 2  
  
**Рекомендации по выполнению**  
• Тип int вмещает в себя не больше десяти цифр (предположим, что у бухгалтера очень маленькие листы бумаги).  
• Проверьте, как программа будет себя вести на отрицательных числах и нуле.  
  
**Что оценивается**  
• В числе ноль программа определяет одну цифру.  
• Количество цифр определяется правильно.

Задание 3. Коллекторы

**Что нужно сделать**  
Напишите робота для коллекторов. В самом начале он спрашивает имя должника и сумму долга, а затем начинает требовать у него погашения долга до тех пор, пока тот не будет погашен полностью. Сумма долга должна уменьшаться, если пользователь ввёл сумму, меньшую чем сумма долга. Если пользователь внёс большую сумму, чем требуется для погашения, выведите сообщение о том, какой у него остаток на счету после такой операции. Обеспечьте контроль ввода.  
  
**Рекомендации по выполнению**  
При полной выплате долга необходимо учитывать остаток на счету, если он имеется.  
  
**Что оценивается**  
• Правильно считается остаток после полного погашения долга.  
• Программа прекращает выполнение после того, как долг полностью погашен.  
• При задании отрицательного долга программа выводит предупреждение и прекращает выполнение.

Задание 4. Счастливый билетик

**Что нужно сделать**  
В старину, когда даже в столице билеты в общественном транспорте выдавали контролёры, существовало поверье: если на билете сумма первых трёх цифр в номере билета равна сумме трёх последних, то это к удаче. Напишите программу, которая получала бы на вход шестизначный номер билета и выводила, счастливый это билет или нет. К примеру, билеты 666 666 и 252 135 — счастливые, а 123 456 — нет.  
  
**Пример выполнения**  
  
Введите номер билета: 123060  
Билет счастливый!  
Введите номер билета: 123040  
Повезёт в следующий раз!  
  
**Рекомендации по выполнению**  
• Программу необходимо сделать так, чтобы можно было увеличить количество цифр в билете без особых изменений в коде.  
• Для получения младшего разряда числа (самой правой цифры) используйте выражение: one\_digit = digits % 10.  
• Используйте циклы для подсчёта суммы цифр в номере билета.  
  
**Что оценивается**  
• Программа корректно определяет, является ли билет счастливым.  
• Для подсчёта суммы цифр в билете используются циклы.

Задание 5. Числа Фибоначчи

**Что нужно сделать**  
Числами Фибоначчи называется следующая последовательность чисел: 1, 1, 2, 3, 5, 8, 13 и так далее (первое число равно 1, второе число равно 1, а каждое следующее равно сумме двух предыдущих). Напишите программу, которая по данному номеру N выводит N-е число этой последовательности (если оно не превосходит 2 147 483 647, конечно же). Обеспечьте контроль ввода.  
  
**Пример выполнения**  
  
Введите номер последовательности: 1  
Число в последовательности: 1  
Введите номер последовательности: 6  
Число в последовательности: 8  
  
**Рекомендации по выполнению**  
• Заметьте, что последовательность Фибоначчи всегда одинаковая. Пользователь вводит только порядковый номер числа из последовательности.  
• Обратите внимание, что условие типа if(value > 2147483647) не имеет смысла.  
  
**Что оценивается**  
• Результатом запроса является число из последовательности Фибоначчи.  
• Число 1 определяется как первое и второе число последовательности.

Задание 6. Вклады

**Что нужно сделать**  
Вклад в банке составляет X рублей. Ежегодно он увеличивается на P процентов, после чего дробная часть копеек отбрасывается. Определите, через сколько лет вклад составит не менее Y рублей. Напишите программу, которая по данным числам X, Y, P определяет, сколько лет пройдёт, прежде чем сумма достигнет значения Y.

**Пример выполнения**  
  
Введите размер вклада: 100  
Введите процентную ставку: 10  
Введите желаемую сумму: 200  
Придётся подождать: 8 лет  
  
**Пример вычисления:**  
  
1 год: 100 + 10% \* 100 = 110  
2 год: 110 + 10% \* 110 = 121  
3 год: 121 + 10% \* 121 = 133  
  
**Рекомендации по выполнению**  
• Расчёты проводятся в целых числах.  
• У вклада есть капитализация процентов (проценты начисляются на весь остаток на счёте).  
  
**Что оценивается**  
Программа аварийно не завершается и не зависает при подсчёте вклада.

Задание 7. Сокращение дроби\* (дополнительное задание)

**Что нужно сделать**  
Пользователь вводит числа M и N. Напишите программу, которая преобразует дробь M/N к несократимому виду и выдаёт получившийся результат. Обеспечьте контроль ввода.  
  
**Пример выполнения**  
  
Введите числитель: 4  
Введите знаменатель: 6  
Результат: 2 / 3  
Введите числитель: 25  
Введите знаменатель: 40  
Результат: 5 / 8  
Введите числитель: −6  
Введите знаменатель: 12  
Результат: −1 / 2  
(При вводе отрицательных чисел в выводе должен быть только один ведущий минус. Неправильный вывод программы: 1/−2, −1/−2.)  
Введите числитель: 896  
Введите знаменатель: 3584  
Результат: 1 / 4  
  
**Рекомендации по выполнению**  
Не забывайте о правилах деления чисел. Программа должна работать корректно и не должна «падать».  
  
**Что оценивается**  
• Алгоритм оптимальный, не бесконечный и не очень большой. Программа не зависает.  
• Правильно определены критерии начала и конца цикла.

Задание 8. Угадай число\* (дополнительное задание)

**Что нужно сделать**  
Напишите программу, которая угадывает число, задуманное пользователем. Число загадывается в диапазоне от 0 до 63. Программа задаёт вопросы вида «Ваше число больше такого-то?» и на основе ответов пользователя («да» или «нет») угадывает число.  
  
**Рекомендации по выполнению**  
• В этом задании можно использовать любой алгоритм поиска, но полный перебор всех 64 значений — самый нежелательный из алгоритмов.  
• Есть алгоритм, который гарантированно даст ответ за семь вопросов.