# Лабораторная работа №1

*Алгоритм. Словесное описание алгоритма.*

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

**1. Алгоритм. Псевдокод.** Решение любой задачи, связанной с вычислениями, вклю - чает в себя выполнение ряда действий в определенном порядке. Процедура решения задачи в виде

1. *действий,* которые надлежит выполнить, и
2. *порядка,* в котором эти действия должны быть выполнены, называется *алгоритмом.*

Словесное описание алгоритма часто называют псевдокодом. *Псевдокод* — это искусст- венный неформальный язык, который помогает программистам разрабатывать алгоритмы. Псев- докод, особенно полезен для разработки алгоритмов, которые преобразуются затем в структурные программы, например, на языке Си. Псевдокод напоминает повседневный язык; он удобен и доста- точно прост, хотя и не является подлинным языком программирования для компьютера.

Программы на псевдокоде на самом деле не выполняются на компьютерах. Скорее они просто помогают программисту «продумывать» программу перед попыткой написать ее на языке программирования, таком, например, как Си. В данной лабораторной работе дается несколько примеров эффективного использования псевдокода при разработке структурированных про- грамм на языке Си.

1. **Управляющие конструкции.** Обычно операторы в программе выполняются один за другим в порядке их записи. Это называется *последовательным выполнением.* Различные опера- торы языка Си, которые будут обсуждать в последующих лабораторных работах, дают програм- мисту возможность указать, что следующий оператор, подлежащий выполнению, может отличать- ся от очередного в последовательности. Это называется *передачей управления.*

Язык Си предоставляет программисту три типа конструкций выбора. В конструкции вы- бора **if** некоторое действие либо выполняется (выбирается), если условие истинно, либо пропус- кается, если это условие ложно. В конструкции выбора **if/else** некоторое действие выполняется, если условие истинно, и выполняется другое действие, если это условие ложно. В конструкции выбора **switch** выполняется одно из набора различных действий в зависимости от значения не- которого выражения.

Конструкция **if** называется *конструкцией с единичным выбором,* поскольку в ней выбира- ется или игнорируется одно действие. Конструкция **if**/**else** называется *конструкцией с двойным выбором,* поскольку в ней выбор происходит между двумя альтернативными действиями. Конст- рукция **switch** называется *конструкцией со множественным выбором,* поскольку в ней выбор происходит из нескольких различных действий.

Язык Си предусматривает три типа конструкций повторения, а именно цикл **while** и цик- лы **do/while** и **for**.

Таким образом, в языке Си имеется только семь управляющих конструкций: последова- тельная, три типа выбора и три типа повторения. Любая программа на языке Си строится путем объединения такого количества управляющих конструкций каждого типа, которое соответствует алгоритму, реализуемому программой. Управляющие конструкции могут присоединяться друг к другу путем соединения точки выхода одной управляющей конструкции с точкой входа после- дующей. Будем называть это *суперпозицией управляющих конструкций.* Существует еще один способ их соединения — метод, называемый *вложением управляющих конструкций.* Таким обра- зом, любая программа на языке Си, может быть построена всего лишь из семи различных типов управляющих конструкций, объединенных одним из двух возможных способов.

**Конструкция выбора if.** Конструкция выбора используется для избрания одного из альтернативных направлений действий. Например, предположим, что проходной балл на экзамене равен 60. Оператор псевдокода

*Если оценка студента больше или равна 60*

*Вывести на экран «Экзамен сдан»*

определяет, является ли условие «оценка студента больше или равна 60» истинным или ложным. Если условие истинно, на экран выводится «Экзамен сдан» и «выполняется» следующий по порядку оператор псевдокода (не забывайте, что псевдокод не является настоящим языком программирования). Если условие ложно, вывод на экран игнорируется и выполняется следующий по порядку оператор псевдокода. Вторая строка конструкции выбора, как правило, записывается с отступом.

Описанный выше условный оператор псевдокода может быть написан на Си как

if (grade >= 60)

printf("Passed\n");

Код на Си близко соответствует псевдокоду. Это является одним из свойств псевдокода, делающих его таким полезным инструментом разработки программ.

*Задание 1: Создать в текстовом процессоре MS Word новый документ.*

*Задание 2: Сохранить созданный документ с именем файла, включающим Вашу фамилию и номер группы и номер лабораторной работы в папке D:\1kursd\номер\_группы\ФИО*

*Задание 3: Оформить в этом файле титульный лист отчета по лабораторной работе. Задание 4: Добавить после титульного листа разрыв страницы (меню Вставка – Разрыв –*

*Новая страница) и на новой странице написать условие задачи из следующего задания.*

*Задание 5: Написать оператор псевдокод следующей задачи: «Присутствуют ли на занятие все студенты группы».*

*Задание 6: Условие задачи разместить по центру страницы и выделить полужирным курсивом синего цвета, размер шрифта 16 пт. Оператор псевдокода необходимо выровнять по левому краю и выделить шрифтом Arial, размер шрифта 14пт*

* 1. **Конструкция выбора if /else.** В конструкции выбора **if** указанное действие выполняется только тогда, когда условие истинно; в противном случае действие пропускается. Конструкция выбора **if/else** дает программисту возможность указать, что в зависимости от того, является ли условие истинным или ложным, должны выполняться различные действия. Например, оператор псевдокода

*Если оценка студента больше или равна 60*

*Вывести на экран «Экзамен сдан»*

*иначе*

*Вывести на экран «Экзамен не сдан»*

выводит на экран «Экзамен сдан», если оценка студента больше или равна 60, либо выводит «Экзамен не сдан», если оценка студента меньше 60. В любом случае после вывода на экран

«выполняется» следующий по порядку оператор псевдокода. Тело для ветви else также записано с отступом.

*Задание 7: Написать оператор псевдокод следующей задачи: «Определить максимальное и минимальное значения из двух различных вещественных чисел».*

*Задание 8: Оформить условие задачи и псевдокод в соответствии с заданием 6.*

Вложенные конструкции **if/else** служат для проверки составных условий, при этом одни конструкции **if/else** помещаются внутри других конструкций **if/else.** Например, следующий оператор псевдокода будет печатать А для экзаменационных оценок, больших или равных 90, В для экзаменационных оценок, больших или равных 80, С для экзаменационных оценок, больших или равных 70, D для экзаменационных оценок, больших или равных 60, и F для всех других оценок.

*Если оценка студента больше или равна 90*

*Вывести «А»*

*иначе*

*Если оценка студента больше или равна 80*

*Вывести «В»*

*иначе*

*Если оценка студента больше или равна 70*

*Вывести «С»*

*иначе*

*Если оценка студента больше или равна 60*

*Вывести «D»*

*иначе*

*Вывести «F»*

*Задание 9: Написать оператор псевдокод следующих задач:*

*А) Определить максимальное и минимальное значения из трех различных вещественных*

*чисел.*

*Б) Найти сумму двух наибольших из тех различных чисел.*

*В) Найти произведение двух наименьших из тех различных чисел.*

*Задание 10: Оформить условие задачи и псевдокод в соответствии с заданием 6.*

1. **Конструкция повторения while.** *Конструкция повторения* позволяет программисту специфицировать многократное выполнение действия до тех пор, пока некоторое условие ос- тается истинным. Оператор псевдокода

*Пока в моем списке покупок еще остаются пункты*

*Сделать следующую покупку и вычеркнуть соответствующий пункт*

описывает повторяющиеся действия, происходящие во время поездки за покупками. Условие «в моем списке покупок еще остаются пункты» может быть истинным или ложным. Если оно истинно, то выполняется действие «Сделать следующую покупку и вычеркнуть со- ответствующий пункт из списка». Это действие будет многократно выполняться до тех пор, пока условие будет оставаться истинным. Оператор(ы), содержащийся в конструкции повто- рения ***while,*** составляет тело этой конструкции.

В конце концов условие становится ложным (когда последний предмет в списке поку- пок приобретен и вычеркнут из списка). В этот момент повторение завершается и выполняет- ся первый оператор псевдокода, следующий за конструкцией повторения.

1. **Формулирование алгоритмов на основе нисходящего пошагового уточнения: повторение, управляемое контрольным значением.** Рассмотрим следующую задачу:

*Разработать программу для подсчета средней оценки в группе, которая при каждом своем запуске будет обрабатывать произвольное число оценок.*

В настоящем примере не дается никаких указаний относительно того, сколько должно быть введено оценок. Программа должна обрабатывать произвольное их число. Каким образом программа сможет определить, когда ввод оценок должен быть прекращен? Каким образом она узнает, когда должна быть вычислена и выведена на экран средняя оценка в группе?

Один из способов решения этой проблемы состоит в том, чтобы использовать для ука- зания на «конец ввода данных» специальное значение, называемое *контрольным значением*. Пользователь вводит оценки до тех пор, пока не будут введены все «правильные» оценки. По- сле этого пользователь вводит контрольное значение, чтобы показать, что была введена по- следняя оценка. Повторение, управляемое контрольным значением, часто называют неопре- деленным повторением, поскольку число повторений неизвестно до начала выполнения цикла.

Очевидно, что контрольное значение должно выбираться таким образом, чтобы его нельзя было спутать с допустимым входным значением. Поскольку экзаменационные оценки обычно являются неотрицательными целыми числами, приемлемым контрольным значением для этой задачи будет -1. Таким образом, при запуске программы подсчета средней оценки в группе может быть обработан поток входных значений, например, 95, 96, 75, 74, 89 и -1. После этого программа вычислит и выведет на экран среднее значение по группе для оценок 95,

96, 75, 74 и 89 (-1 является контрольным значением, поэтому оно не должно учитываться при расчете среднего).

Подход к написанию программы для подсчета средней оценки в группе будет опираться на метод, называемый нисходящим пошаговым уточнением, и который является неотъемле- мой частью разработки хорошо структурированных программ. Начнем с представления псевдокода для верхнего уровня нисходящего процесса:

*Определить среднюю оценку в группе за экзамен*

Верхний уровень представляет собой одно предложение, выражающее общее назначение программы. Как таковой, верхний уровень фактически полностью представляет программу. К сожалению, верхний уровень (как в этом случае) редко несет в себе достаточное количество подробностей, необходимых для написания программы.

Итак, теперь начинается процесс уточнения. Подразделяем верхний уровень на ряд более мелких задач и перечисляем их в том порядке, в котором они должны выполняться. Это приводит к следующему первому уточнению.

*Инициализировать переменные*

*Ввести, просуммировать и подсчитать количество оценок Вычислить и вывести на экран среднюю оценку для группы*

Здесь имеет место последовательная структура — перечисленные шаги должны выполняться по порядку, один за другим.

Для перехода к следующему уровню детализации, то есть ко второму уточнению, необходимо ввести конкретные переменные. Для решения задачи потребуется текущая сумма чисел, счетчик количества обработанных чисел, переменная для приема значения очередной вводимой оценки и переменная, в которой содержится рассчитанное среднее значение. Оператор псевдокода *Инициализировать переменные* может быть уточнен следующим образом:

*Инициализировать итоговую сумму нулем Инициализировать счетчик нулем*

Обратите внимание, что нужно инициализировать только итоговую сумму и счетчик; пе- ременные для рассчитанного среднего значения **average** и ввода пользователя **grade** инициализировать не обязательно, поскольку их значения перезаписываются в результате процесса ввода. Для оператора псевдокода *Ввести, просуммировать и подсчитать количество оценок* потребуется конструкция повторения (т.е. цикл), в которой последовательно вводится каждая оценка. Поскольку не известно заранее количества оценок, которое должно быть обработано, необходимо использовать повторение, управляемое контрольным значением. Пользователь по одной будет вводить допустимые оценки. После ввода последней допустимой оценки пользователь введет контрольное значение. Программа будет осуществлять проверку после ввода каждой оценки и завершит цикл, когда будет введено контрольное значение. Уточнением предыдущего оператора псевдокода тогда будет

*Ввести первую оценку*

*Пока пользователь не ввел контрольного значения Прибавить эту оценку к текущему итогу Прибавить единицу к счетчику оценок*

*Ввести следующую оценку (возможно, контрольное значение)*

В псевдокоде выравниваются все операторы под оператором цикла **while***,* чтобы показать, что все они принадлежат этой конструкции **while***.*

Оператор псевдокода *Вычислить и вывести на экран среднюю оценку для группы* может быть уточнен следующим образом:

*Если счетчик не равен нулю*

*Присвоить переменной average сумму, деленную на счетчик Вывести на экран average*

4

*иначе*

*Вывести на экран «Не было ввода оценок»*

Здесь осуществляется проверка деления на ноль — фатальной ошибки, которая, остав- шись невыявленной, способна привести к отказу в работе программы. Полностью второе уточ- нение имеет следующий вид:

*Инициализировать итоговую сумму нулем Инициализировать счетчик нулем*

*Ввести первую оценку*

*Пока пользователь не ввел контрольного значения Прибавить эту оценку к текущему итогу Прибавить единицу к счетчику оценок*

*Ввести следующую оценку (возможно, контрольное значение)*

*Если счетчик не равен нулю*

*Присвоить переменной average сумму, деленную на счетчик Вывести на экран average*

*иначе*

*Вывести на экран «Не было ввода оценок»*

Несколько пустых строк включаются в псевдокод для повышения его удобочитаемости.

Фактически пустые строки разделяют эти программы на различные этапы их выполнения.

Алгоритм на псевдокоде, приведенный выше, решает задачу о подсчете средней оценки в группе, был разработан после всего лишь двух этапов уточнения. Иногда требуется большее ко- личество этапов.

*Задание 11: Используя метод нисходящим пошаговым уточнением написать псевдокод следующей задачи: «Необходимо определить общую зарплату для каждого из нескольких слу- жащих. Компания платит «обычную зарплату» за первые 40 часов, отработанных каждым служащим, и «полуторную зарплату» за все время, отработанное сверх 40 часов. Вводиться ФИО служащих компании, число часов, отработанных каждым служащим за последнюю неде- лю, и почасовой тариф каждого служащего. Программа должна определить и отобразить на экране его общую зарплату. Количество сотрудников сразу точно неизвестно, для завершения выполнения программы необходимо использовать контрольное значение».*

*Задание 12: Оформить условие задачи и псевдокод в соответствии с заданием 6. Каждый шаг метода нисходящего пошагового уточнения выделить отдельным цветом шрифта.*

1. **Формулирование алгоритмов на основе нисходящего пошагового уточнения: вложенные управляющие структуры.** Рассмотрим еще один пример написания словесного алгоритма решения задачи, используя для этого псевдокод и процесс нисходящего пошагового уточнения. В предыдущем примере управляющие конструкции последовательно помещались одна на другую, подобно тому, как ребенок укладывает кубики. В этом примере демонстрируется един- ственный отличный от предыдущего способ, которым в Си могут соединяться управляющие структуры, а именно *вложение* одной управляющей структуры в другую.

Рассмотрим следующую постановку задачи:

*Колледж предлагает платный курс для подготовки студентов к сдаче государственного экзамена на получение диплома брокера по недвижимости. В прошлом году несколько студентов из прослушавших этот курс сдавали экзамен на получение диплома. Естественно, что колледж хотел бы знать, насколько успешно его студенты сдали этот экзамен. Необходимо написать программу для обобщения результатов экзамена. Есть список этих 10 студентов. Напротив фамилии каждого студента проставлена 1, если студент сдал экзамен, и 2, если он экзамен не сдал.*

Программа должна анализировать результаты экзамена следующим образом:

1. Ввести результат каждого экзамена (т.е. 1 или 2). При каждом очередном запросе ре- зультата экзамена отображать на экране сообщение «Введите результат».

5

1. Подсчитать количество результатов каждого типа.
2. Отобразить краткую сводку результатов экзамена, указав количество студентов, сдавших экзамен, и количество студентов, его не сдавших.
3. Если экзамен сдали более 8 студентов, вывести сообщение «Повысить плату за курс». После тщательного изучения постановки задачи можно сделать следующие наблюдения:
4. Программа должна обработать 10 результатов экзамена. Будет использован цикл, управляемый счетчиком.
5. Каждый результат экзамена представляет собой число, 1 или 2. Всякий раз, когда про- грамма считывает результат экзамена, она должна определять, является ли это число 1 или 2. В алгоритме необходимо проверять на 1. Если число не равно 1, предполагается, что это 2.
6. Используются два счетчика — один для подсчета числа студентов, сдавших экзамен, и другой для подсчета числа студентов, его не сдавших.
7. После того, как программа обработает все результаты, она должна решить, сдали ли экзамен более 8 студентов.

Перейдем к нисходящему пошаговому уточнению алгоритма. Начинаем с представления на псевдокоде верхнего уровня:

*Проанализировать результаты экзамена и решить, должна ли быть повышена плата за курс*

Важно акцентировать внимание на том, что верхний уровень полностью представляет программу

Первым уточнением является:

*Инициализировать переменные*

*Ввести десять экзаменационных оценок и подсчитать число сданных и несданных экзаменов*

*Вывести краткую сводку результатов экзамена и решить, должна ли быть повышена плата за курс*

Хотя здесь также имеем полное представление всей программы, необходимо дальнейшее уточнение. Теперь введем конкретные переменные. Нам потребуются счетчики для регистрации количества сданных и несданных экзаменов, счетчик для управления выполнением цикла и переменная для хранения пользовательского ввода. Оператор псевдокода

*Инициализировать переменные*

может быть уточнен следующим образом:

*Инициализировать нулем количество сданных экзаменов Инициализировать нулем количество несданных экзаменов Инициализировать счетчик студентов единицей*

Обратите внимание, что инициализируются только счетчики и итоговые суммы. Оператор псевдокода

*Ввести десять экзаменационных оценок и подсчитать число сданных и несданных экзаменов*

требует цикла для последовательного ввода результатов каждого экзамена. Здесь известно заранее, что имеется в точности десять результатов экзамена, следовательно, подойдет цикл, управляемый счетчиком. Конструкция с двойным выбором внутри цикла (т.е. вложенная в цикл) будет определять для каждого экзамена, является ли он в результате сданным или несданным, и исходя из этого увеличивать соответствующий счетчик. Уточнением предыдущего оператора псевдокода тогда будет

*Пока счетчик студентов меньше или равен десяти Ввести следующий результат экзамена*

*Если студент сдал экзамен*

*иначе*

*Прибавить единицу к счетчику сданных экзаменов*

*Прибавить единицу к счетчику несданных экзаменов*

*Прибавить единицу к счетчику студентов*

Вставка пустых строк для выделения управляющей структуры **if/else** делает программу бо- лее удобочитаемой. Оператор псевдокода

*Вывести краткую сводку результатов экзамена и решить, должна ли быть по- вышена плата за курс*

может быть уточнен следующим образом:

*Вывести количество сданных экзаменов Вывести количество несданных экзаменов Если экзамен сдали более восьми студентов*

*Вывести «Повысить плату за курс»*

Полностью второе уточнение показано ниже**.** В данном случае для выделения структуры

**while** вставлены пустые строки, что повышает удобочитаемость программы.

*Инициализировать нулем количество сданных экзаменов Инициализировать нулем количество несданных экзаменов Инициализируйте счетчик студентов единицей*

*Пока счетчик студентов меньше или равен десяти Ввести следующий результат экзамена Если студент сдал экзамен*

*Прибавить единицу к счетчику сданных экзаменов*

*иначе*

*Прибавить единицу к счетчику несданных экзаменов*

Си.

*Прибавить единицу к счетчику студентов*

*Вывести количество сданных экзаменов Вывести количество несданных экзаменов Если экзамен сдали более восьми студентов*

*Вывести «Повысить плату за курс»*

Теперь псевдокод является достаточно детализированным для преобразования его в код

*Задание 13: Используя метод нисходящим пошаговым уточнением написать псевдокод следующей задачи: «Известны оценки каждого из учеников класса по физике. Посчитать количе- ство пятерок, количество четверок, количество троек и количество двоек»*

*Задание 12: Оформить условие задачи и псевдокод в соответствии с заданием 6. Каждый шаг метода нисходящего пошагового уточнения выделить отдельным цветом шрифта.*

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

*Задание 1:* Создать новый текстовый документ, который будет содержать выполнение индивидуального задания.

*Задания 2:* Написать псевдокод (словесное описание алгоритма) задачи в соответствии с вариантом. Условие и решение задачи в отчете разместить с новой страницы. Применить раз- личные варианты оформления для условия задачи и псевдокода.

***Вариант 1:*** Известны два расстояния: одно в километрах, другое – в футах (1 фут = 0,45

м). Какое из расстояний меньше, а какое больше?

***Вариант 2:*** Даны радиус круга и сторона квадрата. У какой фигуры площадь больше?

***Вариант 3:*** Даны объемы и массы двух тел из разных материалов. Материал какого из

тел имеет большую плотность?

***Вариант 4:*** Известны сопротивления двух несоединенных друг с другом участков элек- трической цепи т напряжение на каждом из них. По какому участку протека- ет меньший ток?

***Вариант 5:*** Дано натуральное число. Определить является ли оно четным.

***Вариант 6:*** Дано двузначное число. Определить какая из его цифр больше: первая или вторая.

*Задания 3:* Используя метод нисходящим пошаговым уточнением написать псевдокод (словесное описание алгоритма) задачи в соответствии с вариантом (при необходимости использовать повторение, управляемое контрольным значением). Условие и решение задачи в отчете разместить на новой странице. Применить различные варианты оформления для условия задачи и псевдокода на каждом шаге метода нисходящего пошагового уточнения.

***Вариант 1:*** Известна масса каждого груза загруженного в автомобиль. Выяснить не превысила ли общая масса всех грузов грузоподъемность автомобиля.

***Вариант 2:*** Известно число жителей, проживающих в каждом доме улицы. Нумерация домов проведена подряд. Дома с нечетными номерами расположены на од- ной стороне улицы, с четными - на другой. На какой стороне улицы проживает больше жителей?

***Вариант 3:*** Известна стоимость каждого из восьми предметов в двух наборах. Какой из наборов предметов более дешевый?

***Вариант 4:*** Известно количество осадков выпавших за каждый день марта. Верно ли, что общее количество осадков в этот месяц превысило соответствующее количество прошлого года?

***Вариант 5:*** Даны десять целых чисел. Определить является ли их сумма четным числом.

***Вариант 6:*** Три группы студентов, в каждой из которых 20 человек, в сессию сдавали по три экзамена. Определить лучшую по среднему баллу группу.

*Задания 4:* Используя метод нисходящим пошаговым уточнением написать псевдокод (словесное описание алгоритма) задачи в соответствии с вариантом (при необходимости использовать повторение, управляемое контрольным значением). Условие и решение задачи в отчете разместить на новой странице. Применить различные варианты оформления для условия задачи и псевдокода на каждом шаге метода нисходящего пошагового уточнения.

***Примечание:*** Для нахождения минимального/максимального элемента в последовательности при вводе необходимо первоначально принять за минимальный/максимальный элемент первый элемент последовательности. Далее каждый введенный элемент сравнивается с минимальным/максимальным на текущий момент элементом. Если он меньше/больше его, то его значение становиться минимумом/максимумом последовательности на данный момент. Далее считывается новый элемент и сравнение повторяется.

***Вариант 1:*** Известны расстояния от Москвы до нескольких городов. Найти расстояние от Москвы до самого удаленного от нее города, из представленных в списке городов.

***Вариант 2:*** Известны данные о температуре воздуха в течение месяца. Определить, сколько раз дней за месяц была самая низкая температура.

***Вариант 3:*** Известны данные о количестве людей, живущих в квартире №1, в квартире №2 и т.д. В каком числе квартир проживает больше всего жильцов?

***Вариант 4:*** Известны максимальные скорости каждой из 20 марок легковых автомобилей. Определить, какую максимальную скорость имеет самый быстрый автомобиль.

***Вариант 5:*** Известен рост каждого человека из группы. На сколько рост самого высокого из них превышает рост самого низкого?

***Вариант 6:*** Известно число учеников в каждом из 20 классов школы. На сколько числен- ность самого большого (по числу учеников) класса превышает численность са- мого маленького класса?