

Лабораторная работа №9.

Функции пользователя

Приёмы построения алгоритмов

Для создания работающей программы разработчик должен пройти три основных этапа:

проектирование – построение алгоритма решения поставленной задачи;

кодирование – перевод алгоритма на язык программирования;

отладка – проверка работы программы на заранее разработанном наборе тестовых примеров (под тестовым примером подразумевают набор исходных данных и результат, который должен получиться в процессе выполнения разработанной программы).

Существуют разные способы создания алгоритмов. Известна доказанная теорема о том, что любой алгоритм можно построить при помощи трех алгоритмических структур: следования, развилки и цикла. Объединяя различным образом эти структуры можно разработать алгоритм, а по нему составить программу для решения любой логической задачи на компьютере. В этом случае говорят о **структурном программировании**.

Одним из приемов структурного программирования является разбиение (**декомпозиция**) решаемой задачи на логически завершённые подзадачи. Для каждой подзадачи строится так называемый **вспомогательный алгоритм**. В связи с этим можно представить два способа построения алгоритмов:

метод последовательной детализации;

сборочный метод.

Сборочный метод предполагает, что при решении задачи у программиста должна быть библиотека модулей (модуль содержит подпрограмму, которая позволяет решать общие, но небольшие задачи). При решении достаточно сложной задачи алгоритм можно составить из таких модулей. Недостаток метода заключается в том, что для решения конкретной задачи необходимо наличие алгоритмов решённых задач.

Метод последовательной детализации можно разбить на три этапа:

1. Строится основной алгоритм решения общей задачи, т.е. исходная задача разбивается на логически завершённые подзадачи.
2. Для каждой подзадачи определяются входные данные, от которых функционально будет зависеть решение конкретной подзадачи; выходной параметр, который должен быть получен на выходе решения конкретной подзадачи; описывается вспомогательный алгоритм решения этой подзадачи.
3. Каждому вспомогательному алгоритму присваивается уникальное имя, и решение исходной задачи будет строиться как логическая цепочка правильно построенных вспомогательных алгоритмов.

Метод последовательной детализации используется для решения задач большой сложности. При этом к построению алгоритмов можно привлечь целый коллектив программистов.

Понятие функции

Определение функции

В любом языке программирования алгоритм решения подзадачи можно описать в виде отдельной подпрограммы. В языке VBA каждая подпрограмма называется **функцией** – это конструкция языка программирования, связанная с решением конкретной части поставленной задачи. Функция – это самостоятельная единица программы. В языке VBA функции разделяются на встроенные и пользовательские (разработаны самим программистом).

Определение функции состоит из описания заголовка и тела функции. После заголовка функции точка с запятой не ставится. Общий вид функции:

Заголовок функции

```
<имя_функции>(<список_формальных_параметров>) As <тип_возвр_результата>  
    <тело_функции>  
end
```

<тип_возвращаемого_результата> – тип результата работы функции, может быть целочисленным, вещественным, символьным или указателем на любой допустимый тип. Результат работы функции возвращается и записывается сразу после имени функции в ее теле в виде переменной или выражения, допустимого в языке VBA. В описании метода последовательной детализации результат работы функции называется выходным параметром.

<имя_функции> – идентификатор (применительно к методу последовательной детализации – это уникальное имя некоторого вспомогательного алгоритма), при помощи которого функцию можно вызвать для выполнения. Надо стараться задавать такие имена пользовательским функциям, которые не совпадали бы с ключевыми словами языка VBA.

<список_формальных_параметров> определяет типы и имена формальных параметров. В <список_формальных_параметров> параметры разделяются запятыми. С точки зрения построения алгоритмов методом последовательной детализации, формальные параметры – это входные данные, от которых функционально зависит решение конкретной подзадачи. Таким образом, формальные параметры – это переменные, которые позволяют произвести обмен между вызываемой (решает подзадачу) и вызывающей функциями. <список_формальных_параметров> может отсутствовать. В этом случае в заголовке функции после ее имени пишутся «пустые» скобки.

<тело функции> – алгоритм решения подзадачи, включая описание локальных, т.е. нужных только внутри тела функции, переменных, записанный на языке программирования. Операторы тела функции выполняются до первого оператора return. Если в теле функции нет такого оператора (т.е. функция не возвращает никакого результата), то выполняются все операторы до закрывающейся операторной скобки.

```
Function Sum(x as Integer, y as Integer) As Single
```

```
    x=x^2
```

```
    Sum = x+y
```

End Function

```
Z = Sum(1, 5)
D = Sum(Z, 56)
```

Область видимости переменных

Каждая переменная, используемая в теле функции, должна быть объявлена в теле функции, в списке формальных параметров или вне функции. Переменные, используемые в функциях:

глобальные (внешние): объявленные в программе либо как внешние;

локальные (внутренние): объявленные, доступные и используемые только в теле функции;

формальные – переменные, объявленные в списке формальных параметров и доступные только в теле функции.

Объявление переменной вводит её имя в области видимости. Имя переменной может использоваться только в определенной части программы. Для локальной переменной, описанной в теле функции, область видимости начинается с точки объявления до конца тела функции.

Скрытие имен неизбежно при написании больших программ. В этом случае легко не заметить, что имя скрыто и некоторые ошибки, возникающие из-за этого, очень трудно обнаружить. Следовательно, от скрытия имен нужно уходить при написании программ. Использование для глобальных переменных часто употребляемых имен вроде *i* или *x* как раз и приводит к ошибкам, вроде переобъявления переменной.

При объявлении переменная создается, когда встречается её описание, и уничтожается, когда её имя выходит из области видимости. Переменные с глобальными именами создаются и инициализируются только один раз и “живут” до завершения программы.

Параметры функции

Параметры функции определяются списком формальных параметров. Область видимости формальных параметров – тело функции. Имена формальных параметров используются в операторах тела функции. Формальными параметрами могут быть только идентификаторы.

Кроме формальных параметров есть фактические. При вызове функции фактические параметры заменяют формальные, т.е. все формальные параметры в теле функции будут заменены на соответствующие фактические. Этот процесс называется передачей фактических параметров в функцию. Это делается для обмена фактическими данными между вызывающей и вызываемой функцией и получения фактического результата после завершения работы вызываемой функции для передачи его в вызывающую функцию.

Между списками формальных и фактических параметров при вызове функции должно быть соответствие:

1) количества формальных и фактических параметров;

2) последовательности формальных и соответствующих им фактических параметров (первому формальному параметру соответствует первый фактический, второму – второй и т. д.);

3) типов формальных и соответствующих им фактических параметров.

Сопоставление формальных и фактических параметров производится при каждом вызове функции: первому формальному параметру будет сопоставлен первый фактический, второму – второй и т.д. При передаче параметров нарушение соответствия по количеству или типам параметров может привести к серьезным ошибкам, например к нарушению границ переменных, в том числе массивов. Некорректная по смыслу передача фактических параметров может привести к неверному результату, хотя алгоритм, записанный в функции, будет верен.

Список формальных параметров может содержать нуль и более объявлений переменных, разделенных запятыми. При объявлении формальных параметров для каждого элемента списка формальных параметров объявляется имя и тип только этого параметра. Объявление каждого формального параметра имеет форму

`<идентификатор_параметра> As <имя_типа_параметра>`

Здесь `<имя_типа_параметра>` – имя допустимого типа значений, стандартного или определенного пользователем; `<идентификатор_параметра>` – имя формального параметра.

Отсутствие имени типа параметра в списке формальных параметров является синтаксической ошибкой.

Организация вызова функции

Форма вызова функции:

`<имя_функции>(<список_фактических_параметров>);`

Если при описании функции список формальных параметров отсутствовал, то при вызове функции `<список_фактических_параметров>` так же должен отсутствовать. При вызове функции ей передается управление. Опишем организацию вызова функции.

Каждая строка в программе определяется своим адресом. Строка вызываемой функции, в которой вызывается другая функция, также имеет свой адрес, называемый точкой вызова функции. При вызове функции, в ОП организуется структура, которая называется **стеком**.

При вызове функции стек заполняется следующим образом:

1. В стек помещается адрес вызова функции.
2. В стек копируются значения переданных в функцию фактических параметров.
3. Выполняются операторы тела функции.
4. После того как функция завершила свою работу из стека в ячейки ОП, выделенные под переменные, являющиеся фактическими параметрами,

копируются (возвращаются) скопированные туда их же значения при вызове функции.

5. Из стека забирается адрес точки вызова функции, и по этому адресу происходит возврат в вызывающую функцию.

Затем выполняются операторы вызывающей функции, следующие за строкой вызова функции.

Задание. Разработать алгоритм и по нему составить программу с использованием функций пользователя.

1. Распечатать список учеников, фамилии которых начинаются на заданную букву, с указанием даты их рождения и возраста.
2. Распечатать список спортсменов, которые занимаются плаванием с указанием их возраста.
3. Распечатать список учеников, имеющих средний балл по математике и русскому языку выше среднего в классе. Вычислить средний балл учеников класса, если известны оценки каждого ученика по математике, русскому языку и информатике.
4. Распечатать список рабочих бригады, начинающиеся с букв А и С, с указанием их месячной зарплаты и стажа.
5. Распечатать список магазинов, продающих конфеты, выпускаемые заданной кондитерской фабрикой, стоимость которых от 130 до 150 руб. за 1 кг. Указать срок их годности.
6. Распечатать учеников музыкальной школы, которые учатся играть на заданном музыкальном инструменте не менее N лет с указанием их возраста.
7. Распечатать список работников заданного предприятия, чья заработная плата за месяц ниже средней по этому предприятию со стажем более 10 лет, с указанием их фамилии, зарплаты, стажа работы и должности.
8. Распечатать учеников, которые не получили ни одной тройки за последнюю четверть с указанием в каких классах учатся эти ученики и их среднего балла.
9. Распечатать список детей заданного детского сада, которые родились в определенном месяце с указанием их возраста и группы.
10. Распечатать учителей заданной школы, которые преподают математику и информатику с указанием стажа работы и недельной нагрузки.
11. Распечатать учеников, участвовавших в олимпиаде по информатике и заработавших не менее N баллов.
12. учеников заданного класса, которые являются отличниками по итогам года с указанием разницы их среднего балла и среднего балла в классе.
13. Распечатать список самых высоких учеников класса с указанием средней массы мальчиков и среднего роста девочек.

14. По результатам переписи населения распечатать список и общее число жителей, родившихся после N г.
15. абитуриентов, допущенных ко второму экзамену, считая, что при поступлении в университет лица, получившие оценку «неудовлетворительно» на первом экзамене, ко второму экзамену не допускаются.
16. Распечатать список студентов заданной группы, которым назначена стипендия по результатам сессии, используя следующие правила:
 - а) если все оценки 5, назначается повышенная стипендия;
 - б) если все оценки 4 и 5, назначается обычная стипендия;
 - в) если есть оценка 3, стипендия не назначается.
17. Распечатать список студентов заданной группы, у которых средний балл выше указанного с указанием их оценок за последнюю сессию.
18. Распечатать список студентов заданной группы с указанием среднего роста студентов группы, роста самого высокого и самого низкого студента.
19. На аптечном складе хранятся лекарства. Сведения о лекарствах: наименование лекарственного препарата; количество; цена; срок хранения (в месяцах). Распечатать список лекарств, срок хранения которых более 3 месяцев. Указать сколько стоят все препараты, хранящиеся на складе.
20. В столовой предлагается 3 комплексных обеда, состоящих из 5 блюд. Известна стоимость и калорийность каждого блюда. Сколько стоит самый дешевый и самый дорогой обед с указанием блюд, их стоимости и калорийности этих обедов.
21. На торговом складе производится уценка хранящейся продукции по правилу:
 - а) если продукция хранится на складе дольше n месяцев, то она уценивается в 2 раза;
 - б) если срок хранения превысил m ($m < n$) месяцев, но не достиг n , то — в 1,5 раза.Распечатать список товаров с указанием наименования и количества товара, цены товара до и после уценки, срока хранения товара, общей стоимости товаров до и после уценки. Выяснить максимальный и минимальный сроки хранения товаров на складе.
22. N спортсменов–многоборцев принимают участие в соревнованиях по M видам спорта. По каждому виду спорта спортсмен набирает определенное количество очков. Распечатать список спортсменов с указанием очков, которые в сумме набрал каждый спортсмен после окончания соревнований. Вычислить разницу в очках для спортсменов, занявших первое и последнее места.
23. N учеников проходили тестирование, выполнив M тестов по какому-либо предмету. Распечатать список учеников с указанием очков, которые набрал каждый ученик по всем темам. Вычислить разницу между лучшим результатом и средним баллом.
24. Описать переменную служащий. Переменная должна содержать

информацию о имени, фамилии, отчестве служащего, дате рождения, образовании, домашнем адресе, профессии. Распечатать список людей с высшим образованием. Выдать данные о служащем, который имеет заданную профессию.