НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Методические указания для лабораторно-практических занятий
и самостоятельной работы студентов

Новосибирск 2021

УДК 004.42 (07)
ББК 32.973, я 7
О-753

Кафедра Информационных технологий и моделирования

Составитель: А.Ю. Андронов, старший преподаватель кафедры Информационных технологий и моделирования

Рецензент: И.В. Трубчанинова к.э.н., доцент кафедры Учета и финансовых технологий

Основы программирования: методические указания для лабораторно-практических занятий и самостоятельной работы студентов / Новосиб. гос. аграр. ун-т. Фак. ЭиУ; сост.: А.Ю. Андронов – Новосибирск, 2021. – 15 с.

Методические указания для лабораторно-практических занятий и самостоятельной работы студентов по дисциплине «Основы программирования» предназначены для студентов направления подготовки 09.03.03 Прикладная информатика всех форм обучения.

Методические указания утверждены и рекомендованы к изданию учебно-методическим советом факультета экономики и управления (протокол №4 от «28» декабря 2021 г.)

Введение

Методические указания для лабораторно-практических занятий и самостоятельной работы по дисциплине «Основы программирования» предполагает работу с теоретическими темами дисциплины, проявление творческих способностей обучающего. По каждой теме представлено краткое содержание для подготовки к лабораторно-практическим занятиям и тесты для самостоятельной работы обучающегося. Для более подробного ознакомления с каждой темой обучающемуся предлагается использовать библиографический список.

Тема 1. Основы алгоритмизации.

Определение, виды и свойства алгоритмов. Понятие исполнителя. Понятие переменной. Построение линейных и разветвленных алгоритмов .Операция присваивания. Элементарные базовые управляющие структуры: последовательность, ветвление, цикл. Построение алгоритмов.

Решение задач на компьютере основано на понятии алгоритма. Алгоритм – это точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к исходному результату.

Алгоритм означает точное описание некоторого процесса, инструкцию по его выполнению. Разработка алгоритма является сложным и трудоемким процессом. Алгоритмизация – это техника разработки (составления) алгоритма для решения задач на ЭВМ.

Для записи алгоритма решения задачи применяются следующие изобразительные способы их представления:

1. Словесно-формульное описание.
2. Блок-схема (схема графических символов).
3. Алгоритмические языки.
4. Операторные схемы.
5. Псевдокод.

Для записи алгоритма существует общая методика:

1. Каждый алгоритм должен иметь имя, которое раскрывает его смысл.
2. Необходимо обозначить начало и конец алгоритма.
3. Описать входные и выходные данные.
4. Указать команды, которые позволяют выполнять определенные действия над выделенными данными.

Общий вид алгоритма:

- название алгоритма;
- описание данных;
- начало;
- команды;
- конец.

Формульно-словесный способ записи алгоритма характеризуется тем, что описание осуществляется с помощью слов и формул. Содержание последовательности этапов выполнения алгоритмов записывается на естественном профессиональном языке предметной области в произвольной форме.

Графический способ описания алгоритма (блок-схема) получил самое широкое распространение. Для графического описания алгоритмов используются схемы алгоритмов или блочные символы (блоки), которые соединяются между собой линиями связи.

Каждый этап вычислительного процесса представляется геометрическими фигурами (блоками). Они делятся на арифметические или вычислительные (прямоугольник), логические (ромб) и блоки ввода-вывода данных (параллелограмм).Схемы алгоритмов представлены на рисунке 2.

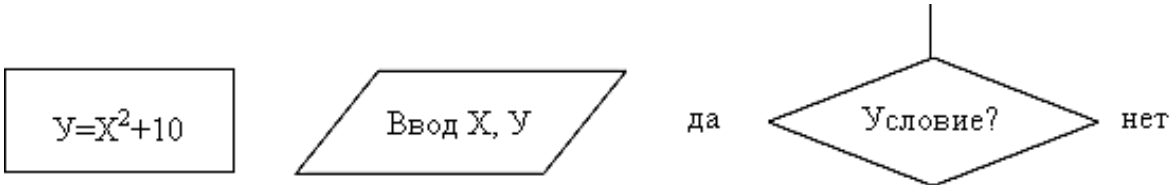


Рисунок 2 – Фигуры в блок схеме

Порядок выполнения этапов указывается стрелками, соединяющими блоки. Геометрические фигуры размещаются сверху вниз и слева на право. Нумерация блоков производится в порядке их размещения в схеме.

Алгоритмические языки – это специальное средство, предназначенное для записи алгоритмов в аналитическом виде. Алгоритмические языки близки к математическим выражениям и к естественным языкам. Каждый алгоритмический язык имеет свой словарь. Алгоритм, записанный на алгоритмическом языке, выполняется по строгим правилам этого конкретного языка.

Операторные схемы алгоритмов. Суть этого способа описания алгоритма заключается в том, что каждый оператор обозначается буквой (например, А – арифметический оператор, Р – логический оператор и т.д.).

Операторы записываются слева направо в последовательности их выполнения, причем, каждый оператор имеет индекс, указывающий порядковый номер оператора. Алгоритм записывается в одну строку в виде последовательности операторов.

Псевдокод – система команд абстрактной машины. Этот способ записи алгоритма с помощью операторов близких к алгоритмическим языкам.

Типы алгоритмических процессов

По структуре выполнения алгоритмы и программы делятся на три вида:

- линейные;
- ветвящиеся;
- циклические.

Линейные вычислительные процессы

Линейный алгоритм (линейная структура) – это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз. Схема представляет собой последовательность блоков, которые располагаются сверху вниз в порядке их выполнения. Первичные и промежуточные данные не оказывают влияния на направление процесса вычисления.

Алгоритмы разветвляющейся структуры

На практике часто встречаются задачи, в которых необходимо выполнять вычисления по одним или другим формулам,в зависимости от первоначальных условий или промежуточных результатов. Данные задачи можно описать с помощью алгоритмов разветвляющейся структуры (рис. 3). В таких алгоритмах выбор направления продолжения вычисления осуществляется по итогам проверки заданного условия. Ветвящиеся процессы описываются оператором IF (условие).

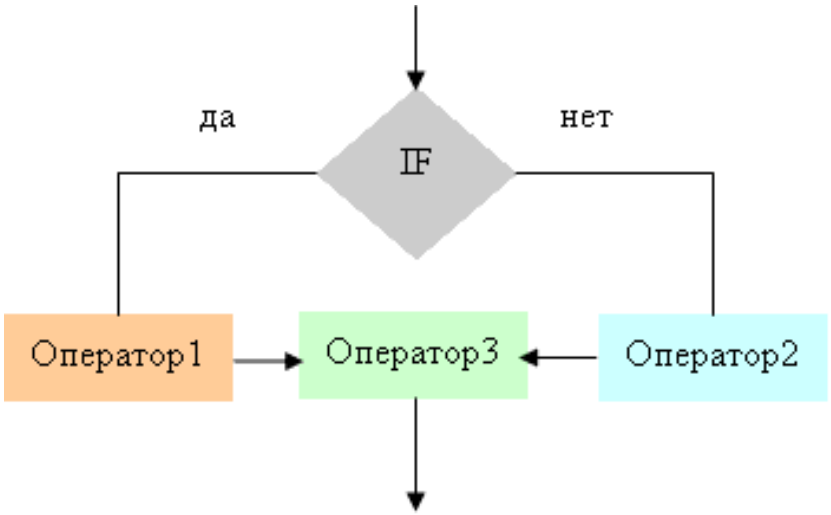
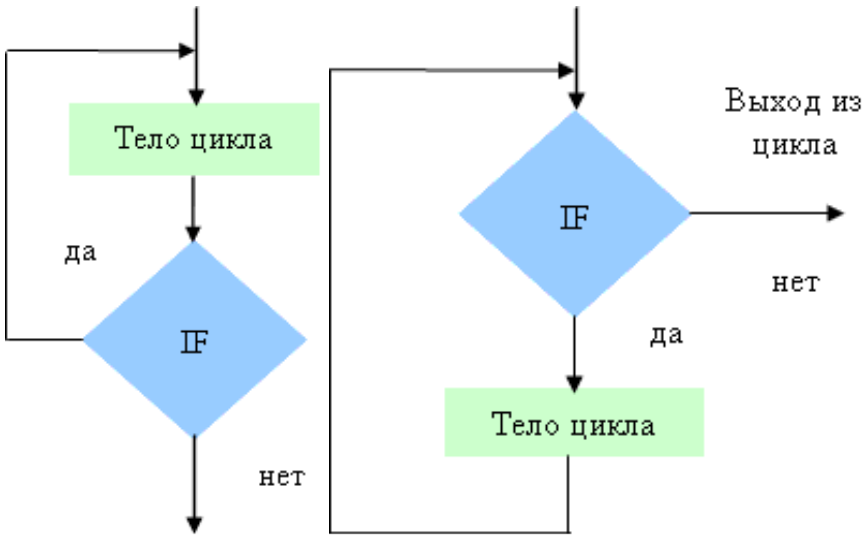


Рисунок 3 – Алгоритм условие

Циклические вычислительные процессы

Для решения многих задач характерно многократное повторение отдельных участков вычислений. Для решения таких задач применяются алгоритмы циклической структуры (циклические алгоритмы). Цикл – последовательность команд, которая повторяется до тех пор, пока не будет выполнено заданное условие. Циклическое описание многократно повторяемых процессов значительно снижает трудоемкость написания программ.

Существуют две схемы циклических вычислительных процессов, (рис. 4,*а, б*).



а) *б)*

Рисунок4 – Цикличные алгоритмы

Особенностью первой схемы (рис. 4, *а*)является то, что проверка условия выхода из цикла проводится до выполнения тела цикла. В том случае, если условие выхода из цикла выполняется, то тело цикла не выполняется ни разу.

Особенностью второй схемы (рис. 4, *б*)является то, что цикл выполняется хоты бы один раз, так как первая проверка условия выхода из цикла осуществляется после того, как тело цикла выполнено.

Существуют циклы с известным числом повторений и итерационные циклы. При итерационном цикле выход из тела цикла, как правило, происходит при достижении заданной точности вычисления.

Форма проведения. Самостоятельная работа по материалам лекций и рекомендованной литературе.

Примеры тестовых заданий для самоконтроля

1. Для вывода значений переменных в блок-схеме используется ...

a. знак равенства;
b. параллелограмм;
c. ромб;
d. треугольник.

2. При изображении блок схемы с помощью овала обозначают ...

a. начало;
b. вывод данных;
c. ввод данных;
d. конец.

Тема 2. Языки программирования.

Назначение, виды и области применения языков программирования. Характеристика и этапы процесса программирования. Классификация языков программирования. Трансляторы постановка задачи, построение модели, разработка алгоритма и проверка его правильности, реализация алгоритма, анализ алгоритма и его сложности, проверка программы, составление документации. Языки программирования высокого уровня. Процедурное и декларативное программирование. Жизненный цикл программы. Среда программирования. Алфавит языка. Структура программы. Программное обеспечение технологии программирования: назначение, виды и решаемые задачи.

Форма проведения. Самостоятельная работа по материалам лекций и рекомендованной литературе.

Примеры заданий для самоконтроля

Написать программу на алгоритмическом языке и на любом языке программирования.

Задание 1.Дано целое число в диапазоне от 0 до 9. Вывести строку – название соответствующей цифры на русском языке (0 – «ноль», 1 – «один», 2 – «два», ...)

Задание 2.Дано целое число в диапазоне от 1 до 5. Вывести строку – словесное описание соответствующей оценки (1 – «плохо», 2 – «неудовлетворительно», 3 – «удовлетворительно», 4 – «хорошо», 5 – «отлично»)

Задание 3.Арифметические действия над числами пронумерованы следующим образом: 1 – сложение, 2 – вычитание, 3 – умножение, 4 – деление. Дан номер действия и два числа A и B (B не равно нулю). Выполнить над числами указанное действие и вывести результат

Тема 3. Типы данных.

Стандартные типы и функции. Математические выражения и операции. Целочисленная арифметика. Характеристика типов. Стандартные порядковые и вещественные типы. Преобразование типов. Целочисленное деление. Логические выражения и операции. Типы и форматы сообщений об ошибках

Форма проведения. Самостоятельная работа по материалам лекций и рекомендованной литературе.

Примеры тестовых заданий для самоконтроля

К базовым алгоритмическим структурам относят:

a. следование, ветвление, цикл;
b. процедура, функция, программа;
c. ветвление и цикл;
d. следование, ветвление, цикл, подпрограмма, программа.

3. Выберите правильную запись выраженияна языке программирования Pascal или C/C++:

a. (a+b*x)/sin(x)
b. a+bx/sin(x)
c. (a+bx)/sin(x)
d. a+bx|sin(x)

Тема 4. Структурное и модульное программирование.

Процедуры и передача параметров. Функции. Глобальные и локальные переменные. Рекурсия.Процедуры и передача параметров. Использование процедур. Глобальные и локальные переменные. Использование функций. Основные принципы структурного программирования. Безусловные конструкции. Модуль CRT. Возможности работы с экраном и звуком. Разработка модуля. Применение пользовательских модулей.Возможности модуля GRAPH. Рисование в текстовом режиме.

Форма проведения. Самостоятельная работа по материалам лекций и рекомендованной литературе.

Примеры заданий для самоконтроля

Задача 1. Алгоритм вычисления значений функций F(w) и Q(w), где w — натуральное число, задан следующими соотношениями:

F(1) = 1; Q(1) = 1;
F(w) = F(w−l) + 2*Q(w−1) при w > 1
Q(w) = Q(w−l) − 2*F(w−1) при w > 1.

Чему равно значение функции F(5)+Q(5)?

Задача 2. Дан рекурсивный алгоритм:

```
procedure F(n: integer);
begin
writeln('*');
if n > 0 then begin
F(n-2);
F(n div 2);
```

```
F(n div 2);
end
end;
```

Сколько символов «звездочка» будет напечатано на экране при выполнении вызова F(5)?

Тема 5. Структуры данных.

Одномерные массивы. Способы заполнения, печать, нахождение суммы элементов. Понятие рефакторинга и оптимизации программного кода. Сортировки массивов данных. Двумерные массивы Основные понятия. Приемы обработки массивов. Анализ элементов массива. Работа с двумя массивами. Применение сортировок данных. Приемы обработки двумерных массивов. Работа с диагональными элементами. Обработка массивов и матриц. Строковый тип. Строковые выражения. Строковые процедуры и функции. Множества. Описание. Операции над множествами.Записи. Описание. Оператор присоединения.Описание файлового типа. Организация доступа к файлам. Стандартные процедуры и функции.Статические и динамические переменные. Указатели и динамическая память.

Форма проведения. Самостоятельная работа по материалам лекций и рекомендованной литературе.

Примеры тестовых заданий для самоконтроля

К простейшим стандартным типам данных относятся типы:

A) double

B) set

C) record

D) file

E) string

К элементарным данным относятся:

A) таблицы

B) данные логического типа

C) данные типа запись

D) списки

E) данные множественного типа

Тема 6. Основы объектно-ориентированное программирование (ООП).

Базовые понятия ООП. Основные принципы ООП. Событийно управляемая модель. Преимущества применения объектно-ориентированного подхода в программировании. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства.

Форма проведения. Самостоятельная работа по материалам лекций и рекомендованной литературе.

Примеры тестовых заданий для самоконтроля

При разработке ПОв первую очередь следует заботиться о?

A. Корректности;
B. интерфейсе пользователя;
C. простоте использования;
D. функциональности.

При разработке сложного ПО основная доля затрат приходится на?

A. Отладку;
B. сопровождение;
C. разработку;
D. создание спецификаций.

Темы контрольных работ

1. Программирование ветвящихся и циклических алгоритмов;
2. Программирование алгоритмов обработки данных в виде массивов;
3. Использование методов, определяемых программистом-пользователем;
4. Обработка данных, организованных в виде массивов и строк;
5. Обработка исключений;
6. Объекты классов, определяемых программистом-пользователем;
7. Отношения между классами;
8. Событийное программирование;
9. Обработка массива объектов классов определяемых пользователем;
10. Классы, производные от библиотечного класса form, и элементы управления;
11. Работа с потоками ввода-вывода, сериализация;
12. Основы компьютерной графики;
13. Работа с коллекциями, язык linq и элементы функционального
14. Программирования;
15. Асинхронные методы и параллельные программы.

Список вопросов к экзамену

1. Определение, виды и свойства алгоритмов.
2. Понятие исполнителя. Понятие переменной. Операция присваивания.
3. Элементарные базовые управляющие структуры.
4. Назначение, виды и области применения языков программирования.
5. Характеристика и этапы процесса программирования.
6. Классификация языков программирования.
7. Трансляторы постановка задачи, построение модели, разработка алгоритма и проверка его правильности, реализация алгоритма, анализ алгоритма и его сложности, проверка программы, составление документации.
8. Языки программирования высокого уровня.
9. Процедурное и декларативное программирование.
10. Жизненный цикл программы.
11. Среда программирования.
12. Алфавит языка. Структура программы.
13. Программное обеспечение технологии программирования: назначение, виды и решаемые задачи.
14. Стандартные типы и функции.
15. Математические выражения и операции.
16. Целочисленная арифметика.

17. Характеристика типов. Преобразование типов.
18. Логические выражения и операции.
19. Типы и форматы сообщений об ошибках
20. Процедуры и передача параметров. Функции.
21. Глобальные и локальные переменные. Рекурсия.
22. Основные принципы структурного программирования.
23. Безусловные конструкции.
24. Модуль CRT. Возможности работы с экраном и звуком.
25. Возможности модуля GRAPH. Рисование в текстовом режиме.
26. Одномерные массивы.
27. Способы заполнения, печать, нахождение суммы элементов.
28. Понятие рефакторинга и оптимизации программного кода.
29. Сортировки массивов данных.
30. Двумерные массивы Основные понятия.
31. Применение сортировок данных. Приемы обработки двумерных массивов.
32. Работа с диагональными элементами.
33. Обработка массивов и матриц.
34. Строковый тип. Строковые выражения. Строковые процедуры и функции.
35. Множества. Описание. Операции над множествами.
36. Записи. Описание. Оператор присоединения.
37. Описание файлового типа. Организация доступа к файлам.
38. Стандартные процедуры и функции.
39. Статические и динамические переменные. Указатели и динамическая память.
40. Основные принципы объектно-ориентированного программирования.
41. Преимущества применения объектно-ориентированного подхода в программировании.
42. Базовые понятия объектно-ориентированного программирования: объект, его свойства и методы, класс, интерфейс.
43. Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.
44. Событийно-управляемая модель программирования.
45. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства.

Библиографический список

Список основной литературы

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. - Москва: ИНФРА-М, 2022. - 343 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст: электронный. - URL: https://znanium.com/catalog/product/1356003

Список дополнительной литературы

1.Затонский, А. В. Программирование и основы алгоритмизации. Теоретические основы и примеры реализации численных методов: учебное пособие / А.В. Затонский, Н.В. Бильфельд. - 2-е изд. - Москва: РИОР: ИНФРА-М, 2020. - 167 с. - (Высшее образование). - DOI: https: //www.dx.doi.org/10.12737/20468. - ISBN 978-5-369-01195-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1077389

Составитель
Андронов Андрей Юрьевич

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Методические указания для лабораторно-практических занятий
и самостоятельной работы студентов

Объем 0,93 уч. – изд. л.

Новосибирский государственный аграрный университет

630039, Новосибирск, ул. Добролюбова, 160

Авторская редакция
Компьютерная верстка А.Ю. Андронов