

ИНФОРМАТИКА

Лекции – С++, среда разработки:

Qt Creator 4.12 Community

Практики (1 – 9 С++):

Среда 10¹⁵-17¹⁵; по индивидуальному графику

Домашние задания: Д31, Д32, Д33.

РК1: Циклические процессы (6 неделя);

РК2: Матрицы (11 неделя);

РК3: Динамические структуры, списки (15 неделя);

Зачет (кафедраальный) по практикам.

Экзамен (результат идет в диплом):

1 – Введение в информатику, С++ - начало;

2 – С++; (модульное программирование)

3 – Задача

Цели и задачи курса

Цель курса информатика – ознакомление с методами и средствами обработки информации и решения задач на ЭВМ. Формирование навыков программирования прикладных задач с использованием языков высокого уровня.

Задачами этого курса является изучение:

- способов представления информации
- основных сведений об ЭВМ
- инструментальных средств программирования
- основ алгоритмизации
- универсального языка программирования C++
- способы тестирования и отладки программ

Часть 1. Основы информатики

1.1 Основные понятия

Информатика – наука, изучающая все аспекты получения, хранения, преобразования, передачи и использования информации.

Под этим понятием объединяют ряд научных направлений, исследующих различные аспекты одного и того же объекта – информации.

1. Теоретическая информатика(теории множеств, алгоритмов, автоматов, сетей, мат. логика, дискретная математика, теория игр и т.д.)
2. Кибернетика(общая теория управления, бионика, автоматика, лингвистика, нейрокибернетика, распознавание образов и т.д.)
3. Программирование(математическое, линейное, динамическое, сетевое, параллельное и т.д.)

Основные понятия(2)

4. Системы искусственного интеллекта.
5. Информационные системы.
6. Вычислительная техника.
7. Информатика в обществе(информационное общество, автоматизация рабочего места, автоматизированные системы обучения).
1. Информатика в природе (информационные процессы в биологических системах).

Базовым понятием для всех направлений является понятие **информации**.

Информация в широком смысле - это самые разнообразные сведения, сообщения, известия, знания и умения (любые виды отражения реально существующего вокруг нас реального мира).

Основные понятия(3)

Информация в узком смысле - это любые сведения, которые являются объектом хранения, передачи и обработки.

Информация передается в виде **информационных сообщений**.

Любое информационное сообщение может иметь произвольную физическую природу (механическую, тепловую, световую, электрическую, акустическую (символ на листе бумаги, световой сигнал, радиоволна и т.д.)

Человек принимает информацию с помощью органов чувств (слух, зрение, осязание, обоняние, вкус и т.д.) и обрабатывает ее в мозгу.

Информация может быть аналоговой и дискретной.

Основные понятия(4)

Аналоговая информация – это информация непрерывная в некотором допустимом диапазоне (температура, давление и т.д.)

Дискретная информация – это информация, которая может принимать только определенные фиксированные значения (датчик вкл. или выкл.).

Разновидностью дискретной информации является **цифровая информация**.

Основной характеристикой информации является **количество информации**.

Так называют числовую характеристику информационного сообщения, которая не зависит от его формы и содержания и характеризует степень неопределенности, которая исчезает после получения сообщения.

Единица количества информации носит наименование бита и представляется одним символом двоичного алфавита.

1.2 Системы счисления

Система счисления – способ представления числовой информации.

Основание системы – число символов, используемое для записи чисел в принятой системе счисления.

Наша традиционная система счисления использует 10 символов (цифр) от 0 до 9 для записи чисел, например 5289.

Количество цифр – количество разрядов числа, которые нумеруются справа налево. В каждом разряде цифры меняются от 0 до 9. При увеличении цифры 9 на 1, нумерация текущего разряда начинается с 0, однако значение старшего разряда увеличивается на 1. Каждый разряд имеет определенный вес 1, 10, 100 и т.д.

Например число 256 состоит из 2 сотен, 5 десятков и 6 единиц: то есть $256 = 2 \cdot 100 + 5 \cdot 10 + 6$

Системы счисления (2)

Так как $100=10^2$ $10=10^1$ $1=10^0$

то

$$256=2*10^2+5*10^1+6*10^0$$

Таким образом, традиционная система основывается на степенях десяти, т.е. любое число записывается по основанию 10, а система называется десятичной. Степень 10 из записи исключается, однако при выполнении операций вес разряда учитывается

3458	4657
+	-
4781	1567
-----	-----
8239	3090

Системы счисления (3)

Другие системы используют и другое количество символов для изображения чисел.

16-ричная система использует для изображения чисел 16 символов. Первые 10 символов совпадают с цифрами 10-тичной системы, а для остальных шести используются буквы латинского алфавита от А до F. Система основывается на степенях 16, т.е. любое число записывается по основанию 16. Например число $3AE_{16}$ это $3 \cdot 16^2 + A \cdot 16^1 + E \cdot 16^0$

Или $3 \cdot 256 + 10 \cdot 16 + 14 = 942_{10}$

$$\begin{array}{r} 1AB \\ + 2E6 \\ \hline 4A1 \end{array}$$

$$\begin{array}{r} 5A8 \\ - 25B \\ \hline 34D \end{array}$$

Системы счисления (4)

8-ричная система использует для изображения чисел 8 символов, которые совпадают с первыми 8 цифрами 10-тичной системы.

Система основывается на степенях 8, т.е. любое число записывается по основанию 8. Например

число 527_8 это $5 \cdot 8^2 + 2 \cdot 8^1 + 7 \cdot 8^0$

Или $5 \cdot 64 + 2 \cdot 8 + 7 = 343_{10}$

В каждом разряде цифры меняются от 0 до 7. При увеличении цифры 7 на 1, разряд обнуляется, однако значение старшего разряда увеличивается на 1. Каждый разряд имеет определенный вес (1,8,64).

$$\begin{array}{r} 342 \\ + 677 \\ \hline 1241 \end{array}$$

$$\begin{array}{r} 456 \\ - 174 \\ \hline 262 \end{array}$$

Системы счисления(5)

2-ричная система использует для изображения чисел всего 2 символа – 0 и 1. При этом любое число может быть представлено как комбинация нулей и единиц.

Система основывается на степенях 2, т.е. любое число записывается по основанию 2.

Например число 1010_2 это $1*2^3+0*2^2+1*2^1+0*2^0$
в десятичной записи это число равно
 $1*8+0*4+1*2+0*1=10_{10}$

При записи числа степени двойки опускаются, однако каждый разряд имеет вес (1,2,4,8...). В каждом разряде цифры меняются от 0 до 1. При переполнении разряда значение текущего разряда обнуляется, а старшего – увеличивается на 1.

$$\begin{array}{r} 10110 \\ + 11010 \\ \hline 110000 \end{array}$$

$$\begin{array}{r} 11011 \\ - 10011 \\ \hline 01000 \end{array}$$

Системы счисления (6)

Двоичная система позволяет работать не только с целыми, но и дробными числами.

Обычную дробь 0.324 можно представить в виде $3/10+2/100+4/1000$ или $3/10^1+2/10^2+4/10^3$, где знаменатели - увеличивающиеся степени 10.

В двоичной дроби в качестве знаменателя используются степени 2.

Так, двоичную дробь 0.101 можно записать $1/2^1+0/2^2+1/2^3$ или $1/2+0/4+1/8$,

что в десятичной системе равно $0.5+0.0+0.125 = 0.625$

Однако, не все дроби можно точно представить в двоичной форме ($1/3$ или $2/5$). Точно представляются только дроби, которые являются степенями 2 ($3/4$ или $5/8$).

При переводе вещественного числа, отдельно переводятся целая и дробная части.

Системы счисления (7)

Так как и 8 и 16 являются степенями 2, то между этими тремя системами существует удобная схема пересчета, которая часто используется в информатике.

$$\begin{array}{ccc}
 0000_2 & 0_{16} & 0101_2 \\
 000_2 & 0_8 & 0101_2
 \end{array}
 \quad
 \begin{array}{ccc}
 5_{16} & & 1111_2 \\
 5_8 & & 1111_2
 \end{array}
 \quad
 \begin{array}{ccc}
 & & F_{16} \\
 & & 17_8
 \end{array}$$

Например

$$\begin{aligned}
 10101_2 &= 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 4 + 1 = 21_{10} \\
 15_{16} &= 1 \cdot 16^1 + 5 \cdot 16^0 = 16 + 5 = 21_{10} \\
 25_8 &= 2 \cdot 8^1 + 5 \cdot 8^0 = 16 + 5 = 21_{10} \\
 10011100 &= 1 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 156_{10}
 \end{aligned}$$

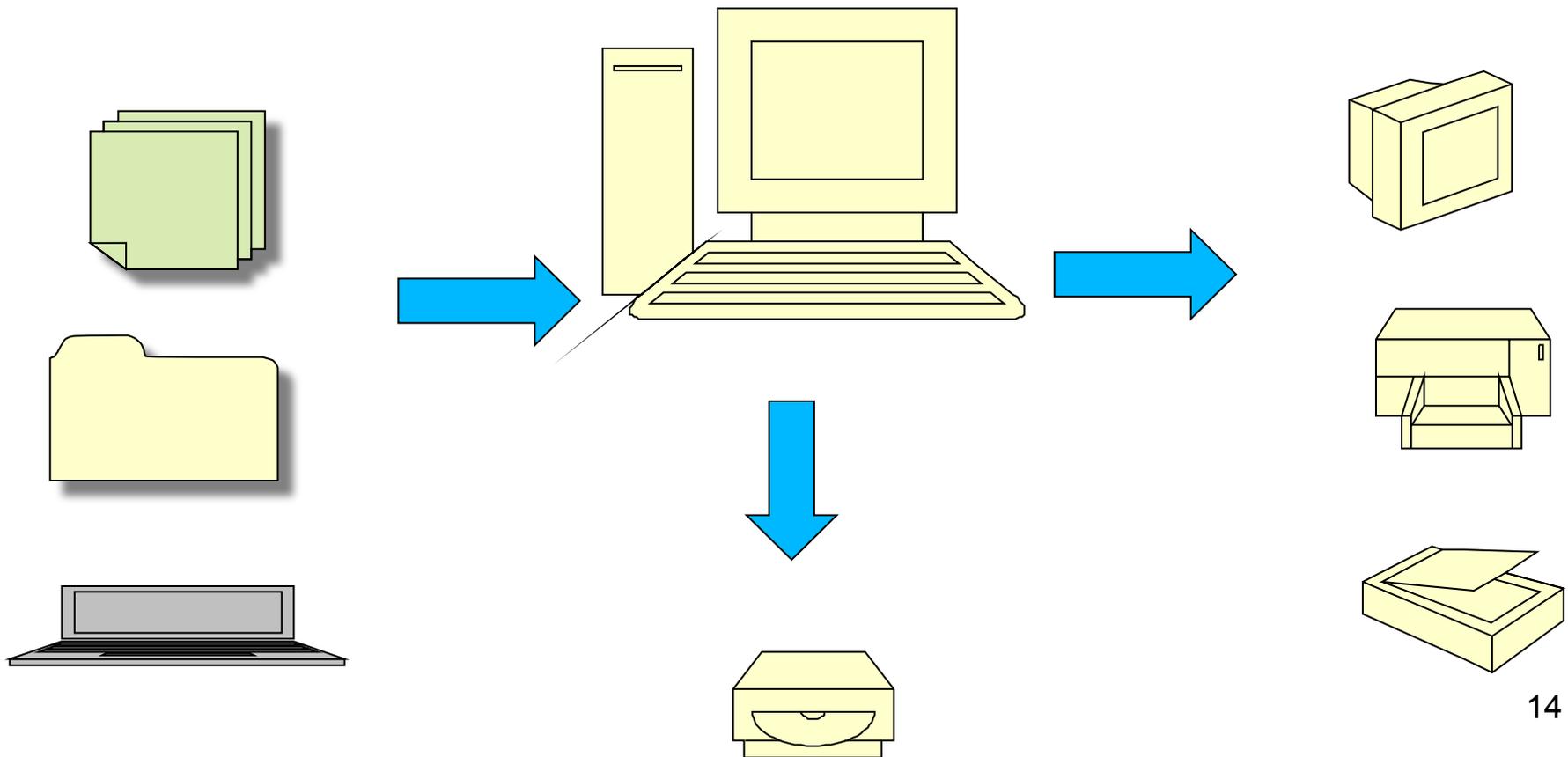
$$\begin{array}{ccc}
 10011100_2 & & \\
 9 & C & \\
 & & 16 \\
 9 \cdot 16 + 12 = & 156_{10} &
 \end{array}$$

$$\begin{array}{ccc}
 010011100_2 & & \\
 2 & 3 & 4 & \\
 & & & 8 \\
 2 \cdot 64 + 3 \cdot 8 + 4 = & 156_{10} &
 \end{array}$$

1.3 Основные сведения об ЭВМ

1.3.1 Архитектура компьютера

Компьютер – это физическая система (устройство или комплекс устройств), предназначенная автоматизации процесса обработки информации.



Архитектура компьютера(2)

Независимо от способа физической организации, любой компьютер можно разделить на шесть функциональных блоков.

- 1. Входной блок.** Получает информацию от различных *входных устройств* для последующей обработки (клавиатура, мышь, речевое устройство..).
- 2. Выходной блок.** Выдает обработанную компьютером информацию на различные *выходные устройства* для дальнейшего использования вне компьютера (принтер, экран, плоттер и т.д.)

Архитектура компьютера(3)

3. **Блок памяти (ОП).** Быстродействующая и относительно малоемкая часть компьютера, Она хранит информацию, которая обрабатывается на компьютере в течении сеанса работы и называется оперативной.
4. **Арифметико – логическое устройство (АЛУ).** Это обрабатывающая часть компьютера. Она отвечает за выполнение различных вычислений. Получает информацию из ОП и возвращает ее в ОП после обработки.
5. **Центральное процессорное устройство (ЦПУ).** Это административная часть компьютера, которая организывает работу компьютера и управляет работой всех его частей.

Архитектура компьютера(4)

6. **Блок вспомогательных запоминающих устройств.** Это долговременные запоминающие устройства большой емкости для хранения информации между сеансами работы компьютера. (жесткие диски, дискеты, флэш память и т.д.).

1.3.2 Способы представления информации в ЭВМ

Вся информация в компьютере представляется в **двоичном виде**.

Наименьшая единица памяти называется **бит**, который может принимать значения 0 и 1. Бит – основной строительный блок памяти, АЛУ и ЦПУ.

Наименьшая адресуемая единица памяти и более удобный ее элемент – **байт**. Байт состоит из 8 бит. Т. к. каждый бит может принять значение 0 и 1, то 8 бит могут представить 256 (2^8) комбинаций из 0 и 1.

Способы представления информации в ЭВМ(2)

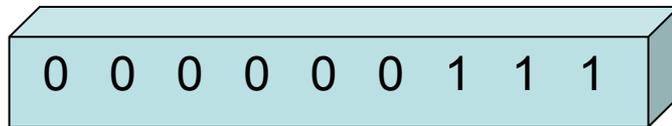
Для удобства обработки, чтения и записи информации байты могут объединяться в слова (2 байта), двойные слова (4 байта) и т.д.

Информация, с которой работает пользователь, бывает числовой, символьной, аудио, видео и т.д.

Для представления числовой информации используются целые и вещественные числа.

Целое число не имеет дробной части (2, -45, 789).

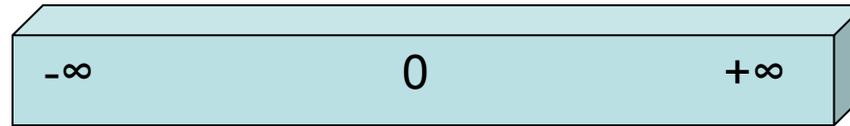
Представив целое число в двоичном виде, его нетрудно разместить в памяти. Например, число 7 – это 111.



$$\begin{array}{ccc} 2^2 & 2^1 & 2^0 \\ 4 & 2 & 1 \\ 4+2+1 & = & 7 \end{array}$$

Способы представления информации в ЭВМ(3)

Целое число



Целые числа могут быть положительными (без знака) и отрицательными (со знаком).

Для хранения знака используется один двоичный разряд (старший). Целые числа являются дискретной информацией и в машине представляются точно.

Вещественные числа – это разновидность аналоговой информации. Включают в себя числа, расположенные между целыми.

В машине такие числа представляются в двоичном виде с определенной точностью. Это связано со схемой размещения и обработки в памяти вещественного числа.

Способы представления информации в ЭВМ(4)



$$+ \quad .314159 \quad \times 10^1 \quad = \quad 3.14159$$

Вещественное число представляется в форме числа с плавающей точкой. Формирование представления такого числа состоит в его разбиении на **дробную часть** и **порядок**, которые затем размещаются в памяти ($7.5 - 0.75 \times 10^1$).

Для размещения чисел в памяти используются двоичные числа и степени двойки вместо степеней десяти. Поэтому точно можно представить только дроби, являющиеся степенями 2.

Однако, такое разбиение дает возможность представить число несколькими способами, например 75×10^{-1} , 7.5×10^0 , 0.75×10^1 .

1.4 Программы и алгоритмы

Основное назначение компьютера – обработка информации, для чего необходимо выполнить определенный набор операций - **программу**.

Программа – набор инструкций, описывающих последовательность действий, приводящих к результату.

Программу можно написать на машинном языке, однако это требует высокой квалификации программиста.

Для возможности написания программы пользователем непрограммистом используют специальные языки называемые языками программирования (Бэйсик, Паскаль, С и т.д.).

Программа на языке программирования преобразуется в машинные команды, которые затем выполняются компьютером.

Программы и алгоритмы (2)

Однако, чтобы составить программу, необходимо хорошо представлять себе, что нужно сделать, чтобы решить какую либо задачу.

Алгоритм – это конечная последовательность четко определенных действий, задающая обработку исходных данных с целью получения нужного результата.

1.4.1 Свойства алгоритмов

1. Массовость (обеспечение функций алгоритма для большой совокупности данных)
2. Дискретность (возможность представить алгоритм в виде отдельных последовательных шагов)
3. Определенность (каждый шаг алгоритма должен быть четко определен и однозначно понятен)

Свойства алгоритмов(2)

4. Результативность (получение нужного результата)
5. Конечность (выполнение алгоритма за конечное число шагов)

1.4.2 Способы представления алгоритма

1. Описательная форма (на естественном языке)
2. Псевдокод (описательная форма с ограниченным числом элементов)
3. Графическая форма (схема алгоритма)
4. Табличная форма (таблицы решений)

Основные конструкции псевдокода

Псевдокод:

1. Следование

...
Действие 1
Действие 2
...

2. Ветвление

...
Если Условие
 то Действие 1
 иначе Действие 2
Все-если
...

3. Цикл-пока

...
Цикл-пока Условие
 Действие
Все-цикл
...

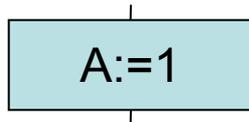
Схемы алгоритмов

Обозначения ГОСТ 19.701 – 90

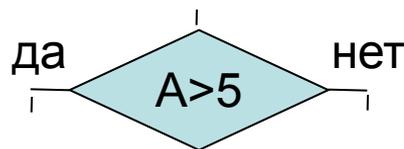
1. Терминатор
(начало/конец)



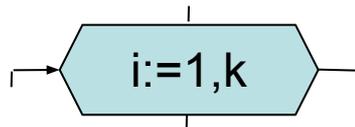
2. Процесс
(вычисления)



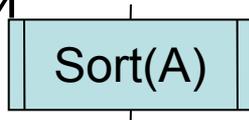
3. Анализ
(проверка)



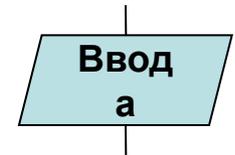
4. Модификатор
(автоматическое изменение)



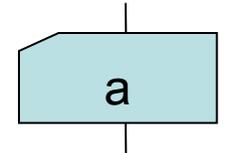
5. Предопределенный процесс
(подпрограмма)



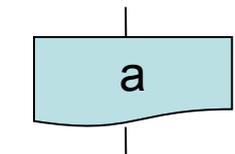
6. Ввод/вывод данных



7. Ввод с перфокарт



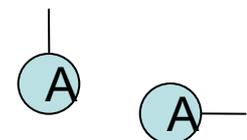
8. Вывод на принтер



9. Комментарий



10. Соединитель



Правила выполнения схем алгоритмов

Схемы алгоритмов должны быть выполнены аккуратно, желательно с применением карандаша и линейки или графических редакторов на компьютере.

Стрелки на линиях, идущих **сверху вниз** и **слева направо**, т. е. в направлении нашего письма **не ставят**, чтобы не затенять схему.

Если линия – ломанная, и направление ее хотя бы в одном сегменте не совпадает со стандартными, то стрелка ставится **в конце линии**, перед блоком, в который она входит.

Если схема не уместится на странице или линии многократно пересекаются, то линии разрывают. При этом один соединитель ставится в месте разрыва, второй – в месте продолжения линии. Оба соединителя помечаются одной и той же буквой или цифрой.

Для простоты чтения схемы ее начало должно быть сверху, а конец – снизу. При этом количество изгибов, пересечений и обратных направлений линий должно быть минимальным.

Таблицы решений

Таблица составляется следующим образом.

В столбик выписываются все условия, от которых зависят дальнейшие вычисления, а по горизонтали - все случаи для вычислений.

На пересечении каждого столбца и строки ставят букву Y, если для данного решения данное условие должно выполняться, букву N, если данное условие обязательно должно не выполняться, и прочерк, если исход сравнения не важен.

Например, для алгоритма вычисления корней квадратного уравнения можно составить следующую таблицу:

Таблицы решений(2)

	Нет корней	$x = -b / 2a$	$x = \pm(-b \pm \sqrt{D}) / 2a$
$D < 0$	Y	N	N
$D = 0$	N	Y	N
$D > 0$	N	N	Y

Иногда, составленная таблица может иметь довольно сложный вид. Рассмотрим, например, таблицу:

	P1	P2	P3	P4
Условие 1	Y	-	N	Y
Условие 2	N	Y	N	N
Условие 3	Y	-	-	N

Таблицы решений(3)

Если строго придерживаться заданного порядка проверки условий, то получится довольно сложный алгоритм и его построение вызывает определенные трудности.

Но этот алгоритм можно значительно упростить, если в таблице поменять местами проверяемые условия, а также для удобства построения алгоритма поменять местами столбцы таблицы. Если преобразовать таблицу следующим образом:

	P1	P4	P3	P2
Условие 2	N	N	N	Y
Условие 1	Y	Y	N	-
Условие 3	Y	N	-	-

Часть 2. Основы алгоритмизации и процедурное программирование

Введение. Этапы создания ПО

1. **Постановка задачи** – неформальное описание задачи
2. **Анализ и уточнение требований** – формальная постановка задачи и выбор метода решения
3. **Проектирование** – разработка структуры ПО, выбор структур данных, разработка алгоритмов, определение особенностей взаимодействия с программной средой
4. **Реализация** – составление программ, тестирование и отладка
5. **Модификация** – выпуск новых версий

Пример разработки программы

1. **Постановка задачи:** Разработать программу, которая определяет наибольший общий делитель двух целых чисел.

2. **Анализ и уточнение требований:**

1) *Функциональные требования*

исходные данные: a, b – натуральные числа; $0 < a, b < ?$;

результат: x – натуральное число, такое, что

$$x = \max \{y_i / i = \overline{1, n}\}, \text{ где } ((a \bmod y_i) = 0) \ \& \ ((b \bmod y_i) = 0)$$

Метод решения:

а) найти делители $Y = \{y_i\}$ и определить $x = \max \{Y\}$;

б) метод Евклида

Пример 1:

a	b
24	18
6	18
6	12
6 =	6

Пример 2:

a	b
3	4
3	1
2	1
1	= 1

Пример разработки программы (2)

2) *Эксплуатационные требования:*

- а) процессор – не ниже Pentium;
- б) операционная система – Windows XP (консольный режим);
- в) предусмотреть запрос на ввод данных с клавиатуры;
- г) результаты вывести на экран дисплея.

3) *Технологические требования:*

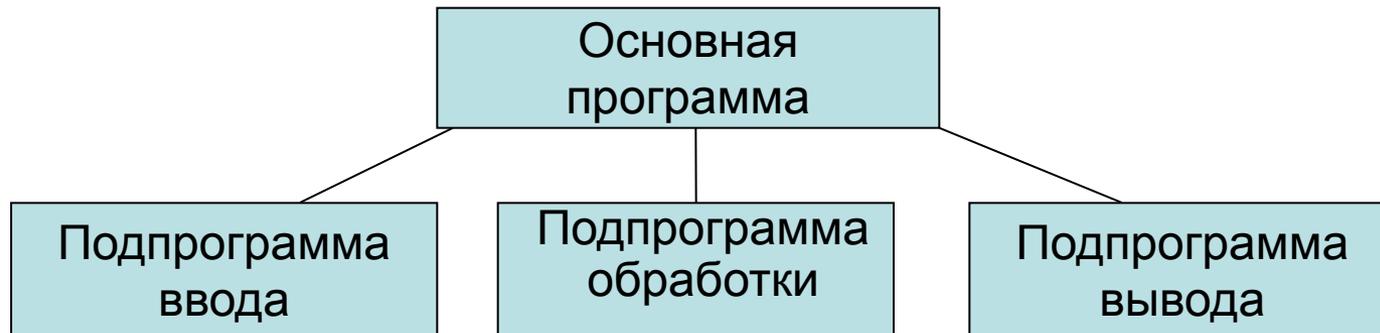
- а) язык программирования: C++;
- б) среда программирования: Microsoft Visual Studio .Net 2003;
- в) технология: структурный подход.

Пример разработки программы(3)

3. Проектирование

Виды проектной документации:

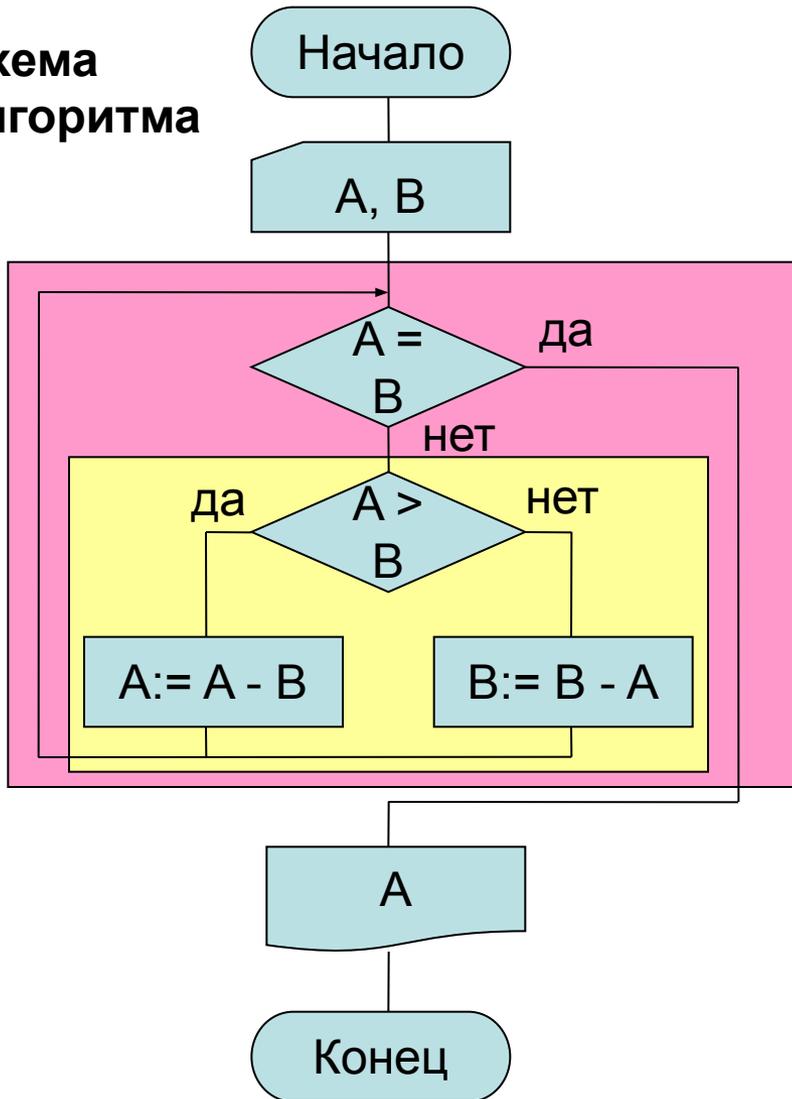
Структурная схема ПО – показывает взаимодействие по управлению основной программы и подпрограмм.



Алгоритм основной программы и подпрограмм в соответствии с выбранным способом представления

Пример разработки программы (4)

Схема алгоритма



Алгоритм на псевдокоде

Начало

Ввести A, B

Цикл-пока $A \neq B$

Если $A > B$

то $A := A - B$

иначе $B := B - A$

Все-если

Все-цикл

Вывести A

Конец

Схема процесса подготовки программы

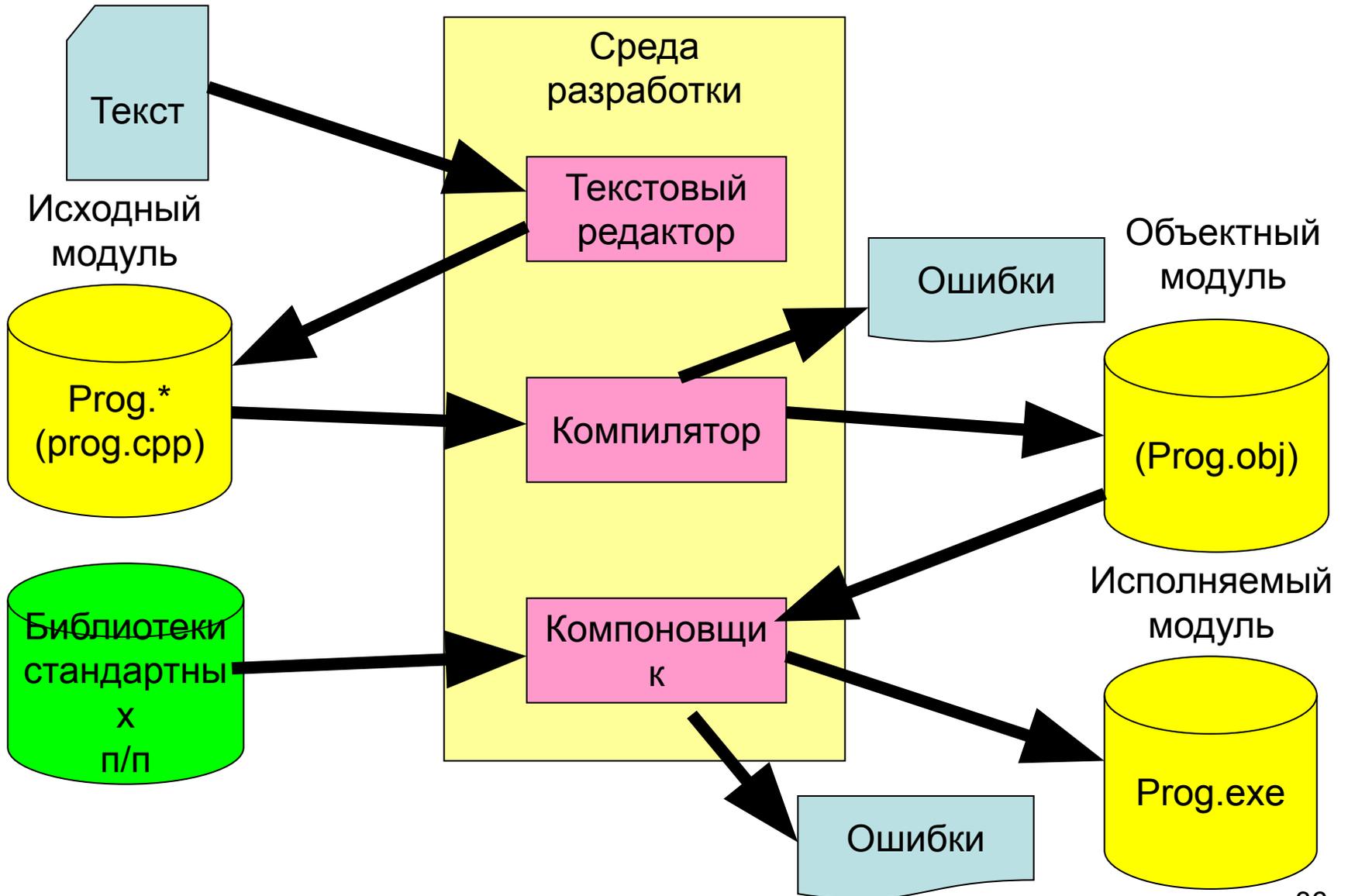


Схема процесса отладки и выполнения

