

ОБЗОР ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Лекция №1

ПОЧЕМУ НАДО САМОМУ ПИСАТЬ ПРОГРАММЫ?

- Стандартные программы и библиотеки программ могут не учитывать особенности решаемой задачи
- Стандартные программы вносят избыточность в программный код
 - Часть стандартного программного кода закрыта
 - Внесение изменений в чужой программный код влечет трудно обнаружимые ошибки
- Платный доступ к некоторым стандартным программам и библиотекам
- *Самостоятельное написание программ – один из путей самореализации*



ЭТАПЫ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- Машинные коды
- Языки низкого уровня
 - Ассемблеры (50-60е годы)
- Языки высокого уровня
 - Фортран (60е годы)
 - Бэйсик (70е годы)
 - Паскаль (70е годы)
 - Си (70-80е годы)
 - Си++ (80е годы)
- Графические языки
 - G
- Языки для сетевых приложений (HTML, XML, CSS, PHP, Java Script и т.д.)

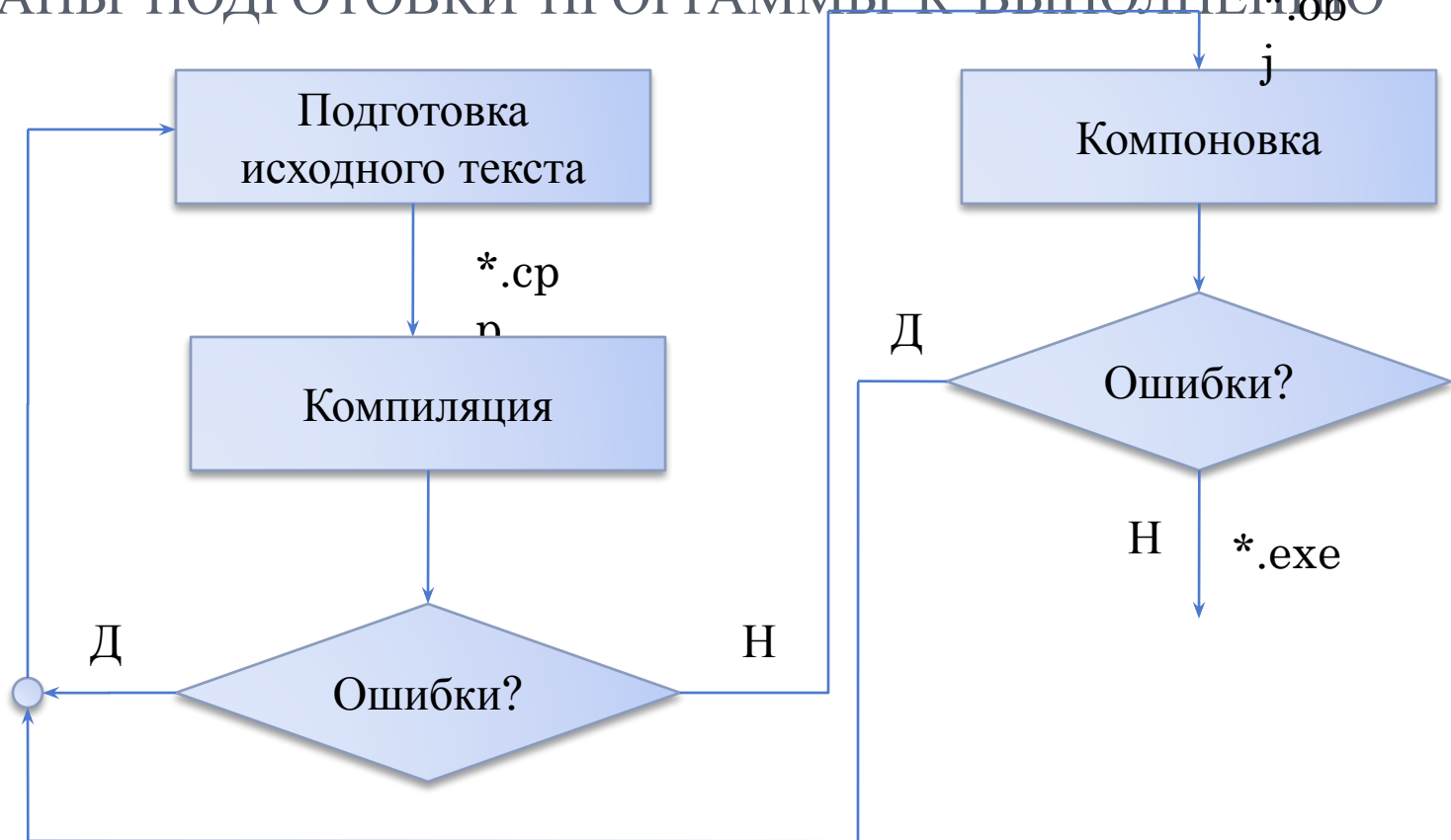


СВОЙСТВА ЯЗЫКА СИ (СИ++)

- Логическая завершенность (последовательность, условие, цикл)
- Модульный принцип построения программы
- Ограниченный набор операций и операторов
- Большой набор стандартных библиотек (ввод-вывод, математика, работа со строками и т.д.)
- Управление памятью с помощью указателей (выделение, освобождение, «перемещение»)
- Наличие операций, совпадающих с командами Ассемблера (инкремент, декремент, управление битами)
- Преобразование исходного текста в машинный код путем компиляции



ЭТАПЫ ПОДГОТОВКИ ПРОГРАММЫ К ВЫПОЛНЕНИЮ



ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА

- Отдельно редактор, компилятор и компоновщик
 - Дешевизна
 - Неудобство отладки
- Интегрированная среда программирования (например, MS Visual Studio 2005 и выше)
 - Текстовый редактор с подсветкой служебных слов
 - Объединение нескольких файлов в проект
 - Отладочные средства (контрольные точки, пошаговая отладка, просмотр переменных)
- Интегрированная среда программирования специализированных устройств (микроконтроллеров, сигнальных процессоров)
 - Преобразование *.cpp -> *.asm



ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

- Структурное программирование
 - Представление программы в виде множества последовательностей, ветвлений и циклов
- Модульное программирование
 - Представление программы в виде множества функций
 - Каждая функция – логически завершённый модуль с несколькими входами и одним выходом
- Объектно-ориентированное программирование
 - Объединение элементов данных и действий над ними в классы



ПРОСТЕЙШАЯ ПРОГРАММА НА СИ

```
/*Моя первая программа hello.cpp*/  
/*Включение в текст заголовочных файлов*/  
#include <stdio.h> //ввод-вывод  
#include <conio.h> //управление вводом-выводом  
/*Главная функция - обязательная*/  
void main(void)//заголовок  
{//тело  
    printf(“Hello, World!\n ”); //вывод на экран сообщения  
    getch();//ожидание нажатия клавиши  
}
```



СВОЙСТВА ПРОГРАММЫ НА СИ

- Программа на Си – совокупность функций
- Функция `main` – обязательная
- Функция состоит из заголовка и тела, не отделимых друг от друга
- Тело выделяется { }
- Разделитель между законченными действиями ;



ЭЛЕМЕНТЫ ЯЗЫКА СИ

- Типы (форматы, способы представления) данных
- Переменные
- Константы
- Операции
- Операторы
- Функции
- Директивы препроцессора (предварительной обработки), начинаются с #



ТИП ДАННЫХ

- Соглашение о способе хранения данных в памяти
 - Целочисленный (int)
 - Вещественный (float)
 - Вещественный с двойной точностью (double)
 - Символьный (char)



КОНСТАНТА

- Объект, который не изменяется в течение работы программы
 - Целочисленная
 - Десятичная (1; 32000; -5)
 - Восьмеричная (03; 032; 0100)
 - Шестнадцатеричная (0x10; 0x2A; 0x1FE)
 - Вещественная
 - Обычная (2.6; -0.01; 3.14)
 - Экспоненциальная (1.3E-3; 2.4E5; .5E4)
 - Символьная
 - Печатаемая ('A'; 'd'; '+')
 - Управляющая ('\n', '\t', '\\')
 - Строковая ("ABCabc123?!+\n")



ПЕРЕМЕННАЯ

- Поименованная область памяти, которая может изменяться в процессе работы программы
 - тип данных
 - имя переменной
 - начальное значение (необязательно)
- Примеры определения переменных
 - `int a=3,b;`
 - `float b=2.68;`
 - `char c, d='A';`
- Использование переменных ***возможно только после их определения***



ОПЕРАЦИИ

- Присваивание (=)
- Арифметические (+ - * / %)
- Инкремент, декремент (++ , --)
- Отношения (> < <= >= == !=)
- Логические (&& || !) – в условных операторах
- Адресные (& *) – при работе с указателями
- Поразрядные (& | ~ ^)
- Обращение к полям структур (. ->)

- Пример
 - $c=(a+b)*2-c/3;$



ОПЕРАТОРЫ

- ; пустой оператор-разделитель
- условный оператор if
- параметрический цикл for
- цикл с предусловием while
- цикл с постусловием do ... while
- оператор-переключатель switch
- операторы для работы с динамическими переменными new и delete



КОМАНДЫ ПРЕПРОЦЕССОРА

□ Включение файлов

- `#include<stdio.h>`//стандартный файл
- `#include "c:\prog\my.h"`//пользовательский файл

□ Определение констант

- `#define A 1024`
- `#define STR "Hello"`
- `#define B 2E-3`



ЗАКЛЮЧЕНИЕ

- Язык Си – средства структурного и модульного программирования
- Язык СИ позволяет реализовать программу любой степени сложности с помощью последовательностей, условного ветвления и циклов
- Программа на Си – совокупность модулей-функций с входными и выходными параметрами

