

Понятие программы

Лекция 1

План лекции

- Информация об этом курсе
- Понятие программы
- Этапы создания программ

Информация об этом курсе

- Петров Евгений Сергеевич
- Программа курса
http://nsu.ru/rs/mw/link/Media:/30785/Б3.Б.4_Программирование.pdf
- Лекция + семинар + практика каждую учебную неделю
- 1й семестр
 - 16 учебных недель
 - 1 потоковая контрольная работа
 - Дифференцированный зачёт
- 2й семестр
 - 16 учебных недель
 - 1 потоковая контрольная работа
 - Экзамен

Информация об этом курсе

- Денис, Ритчи «Язык программирования Си», 3-е издание
- Кормен и ещё три автора «Алгоритмы: построение и анализ», любое издание
- Макконелл «Совершенный код», любое издание
- Чурина, Цикоза методическое пособие в 3-х частях к курсу «Методы программирования»
- Полный список литературы см. в программе курса

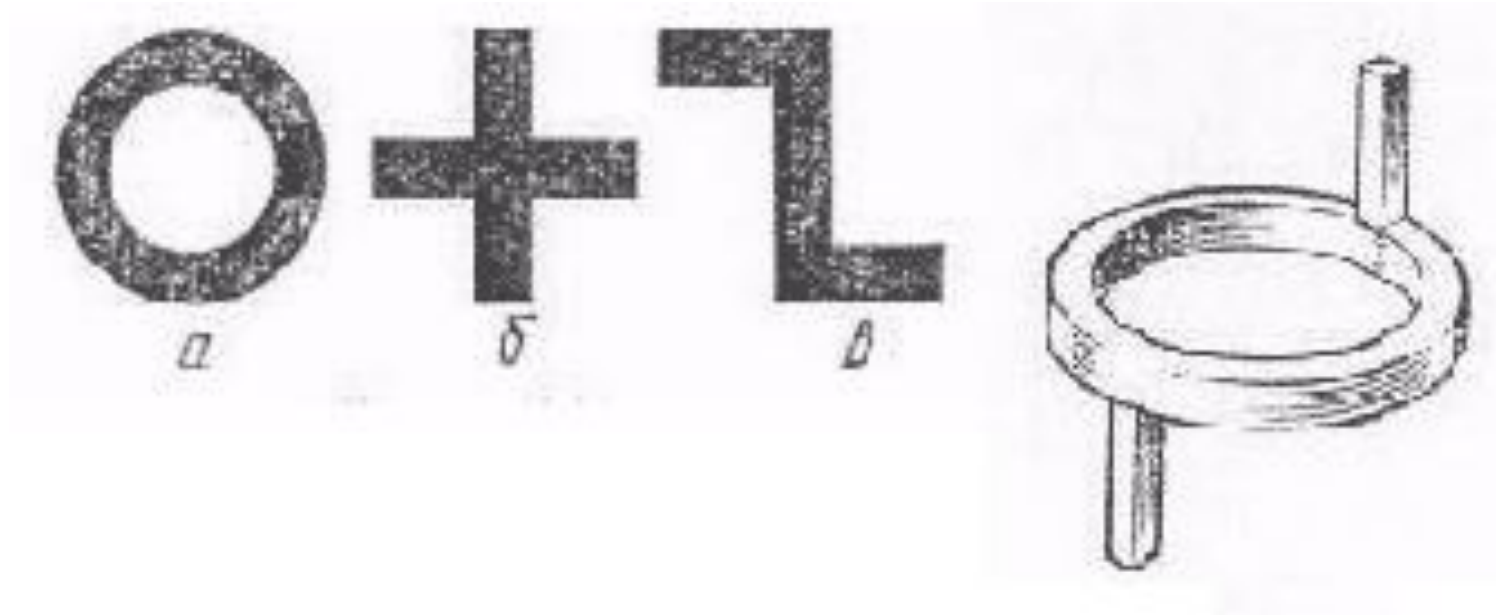
Понятие программы

- Программа – это данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма. (ГОСТ 19781—90)
- Программа – это представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определённого результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения. (ГК РФ)

Понятие программы

- Программа – это размещённые в оперативной памяти компьютера данные и машинные инструкции, исполняемые процессором для достижения некоторой цели. (Википедия)

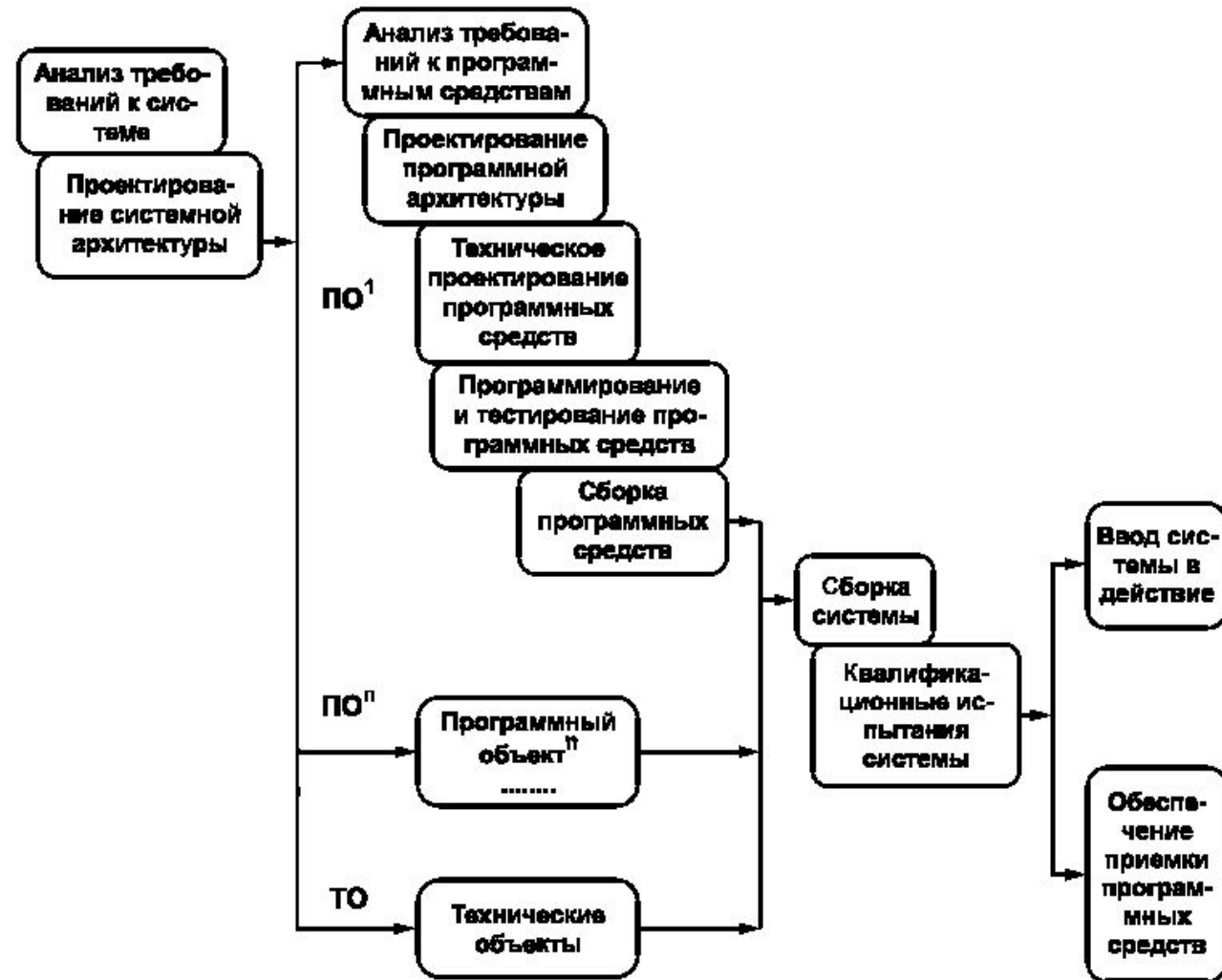
Понятие программы



Этапы создания программ

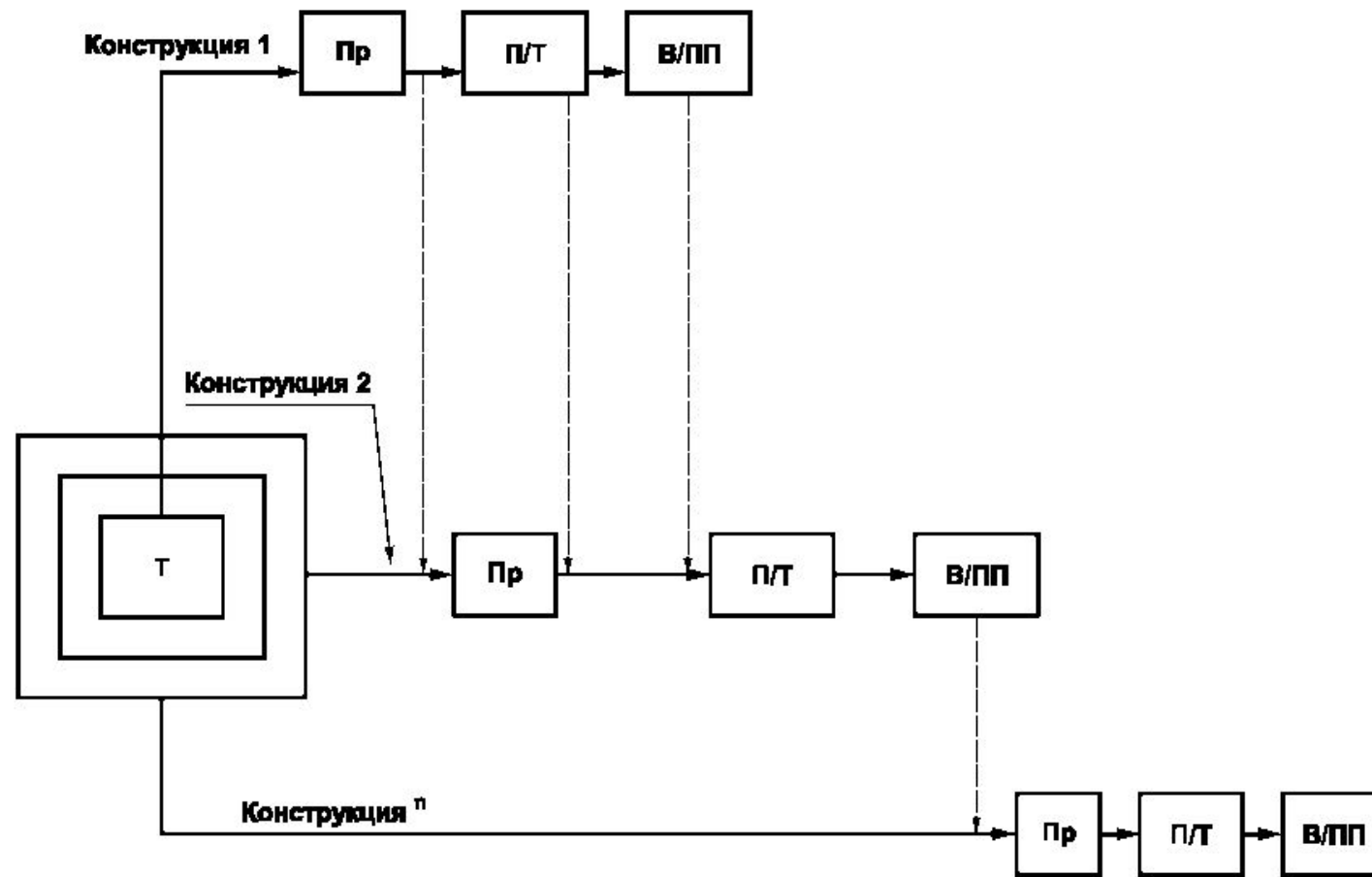
- Накопление требований, работа с заказчиком
- Проектирование – процедурная декомпозиция, ОО, др.
- Внутреннее и внешнее документирование
- Разработка
 - Написание исходного кода
 - Компиляция исходного кода
 - Сборка
 - Отладка
 - Оптимизация
 - Тестирование
- Сдача в эксплуатацию (релиз)
- Сопровождение

Каскадный подход



ПО - программный объект;
ТО - технический объект

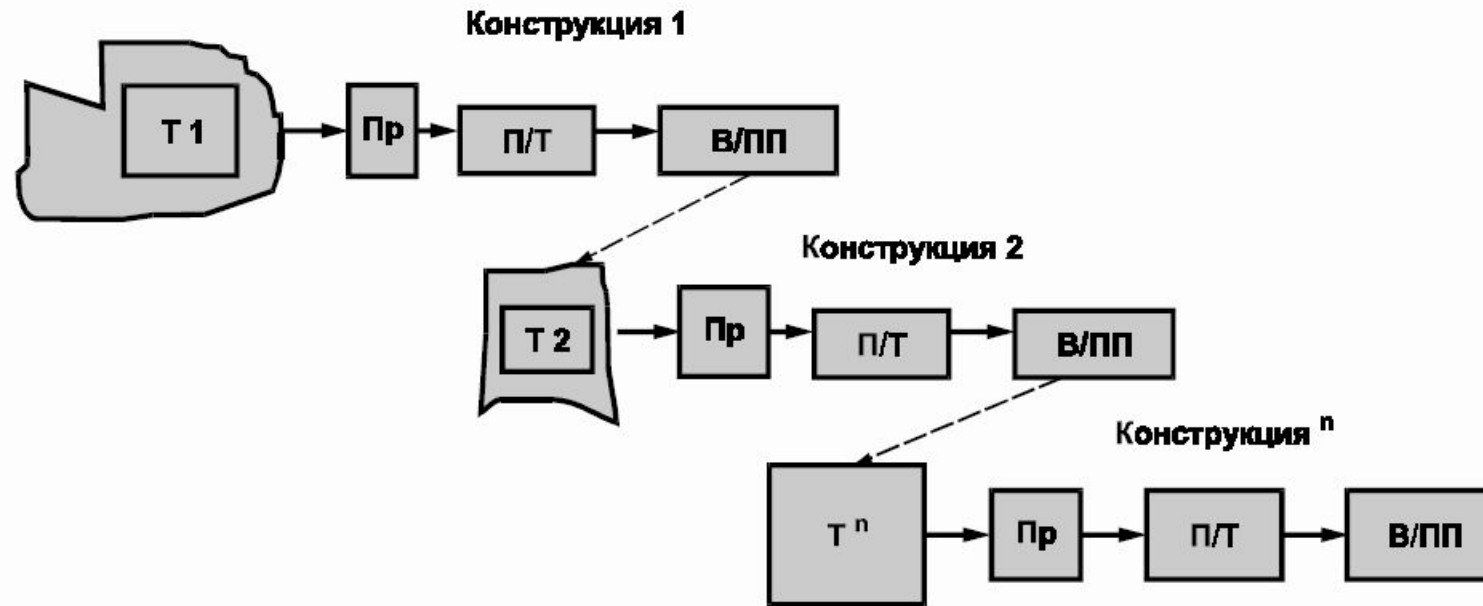
Инкрементальный подход



→ Возможный информационный поток

Т - требования;
Пр - проектирование;
П/Т - программирование и тестирование;
В/ПП - ввод в действие и обеспечения приемки

Эволюционный подход



-----> Информационный поток (уточненный)

Т - требования;

Пр - проект;

П/Т - программирование и тестирование;

В/ПП - ввод в действие и поддержка приемки

Этапы создания программ

	Каскад	Инкремент	Эволюция
Создание программы	Один проход	Итерации	Итерации
Размер программы	Ограниченный	Любой	Любой
Изменение функциональности	Нет	Частично	Без ограничений
Качество программы	Высокое	Ограниченное	Ограниченное
Требования к квалификации разработчиков	Низкие	Высокие	Высокие
Примеры технологий	Плановая	Extreme Programming (XP) Feature-Driven Development (FDD) Agile Programming	

Написание исходного кода

- Выбор языка программирования
- Следование стандарту языка
- Стив Макконнелл Совершенный код (Steve McConnell Code Complete)
 - Следование системе правил
 - Именованние типов, переменных, констант, функций, файлов
 - Деление кода на функции, файлы, компоненты
 - Форматирование и комментирование кода
 - Минимальное дублирование и максимальное переиспользование кода
 - Типовые решения типовых задач
 - Рефакторинг кода (code refactoring)
 - Ревью изменений (code review)
 - ...

Написание исходного кода – венгерская НОТАЦИЯ

Чарльз Симони (Simonyi Károly) р.
1948 Будапешт

Разработка MS DOS

Космический
турист 2005 и 2007



Префикс	Сокращение от	Смысл	Пример
s	string	строка	sClientName
sz	zero-terminated string	строка, ограниченная нулевым символом	szClientName
n, i	int	целочисленная переменная	nSize, iSize
l	long	длинное целое	lAmount
b	boolean	булева переменная	blsEmpty
a	array	массив	aDimensions
t, dt	time, datetime	время, дата и время	tDelivery, dtDelivery
p	pointer	указатель	pBox
lp	long pointer	двойной (дальний) указатель	lpBox
r	reference	ссылка	rBoxes
h	handle	дескриптор	hWindow
m_	member	переменная-член	m_sAddress
g_	global	глобальная переменная	g_nSpeed
C	class	класс	CString
T	type	тип	TObject
I	interface	интерфейс	IDispatch
v	void	отсутствие типа	vReserved

Компиляция исходного кода

- Файлы с исходным кодом называются *единицами компиляции*
- Результатом компиляции является файл с объектным кодом
- Если изменения в файле Ф1 *могут* нарушить логику работы кода в файле Ф2, то Ф2 зависит от Ф1
- Системы компиляции умеют автоматически учитывать *некоторые* зависимости между файлами
 - GNU make, MS nmake, scons, ...
- За учёт всех зависимостей отвечает программист

Компиляция исходного кода

- Стадии работы компилятора
 - Формирование *лексем* (след. лекция)
 - Синтаксический анализ
 - Семантический анализ
 - Оптимизация
 - Генерация кода

Сборка (линковка)

- Различают три вида сборки
 - Сборка статической библиотеки
 - Сборка динамической библиотеки
 - Сборка исполняемого файла

Сборка статической библиотеки

- Вход: объектные файлы
- Выход: архив, содержащий эти файлы
- Статическая библиотека – средство группирования логически связанных объектных файлов

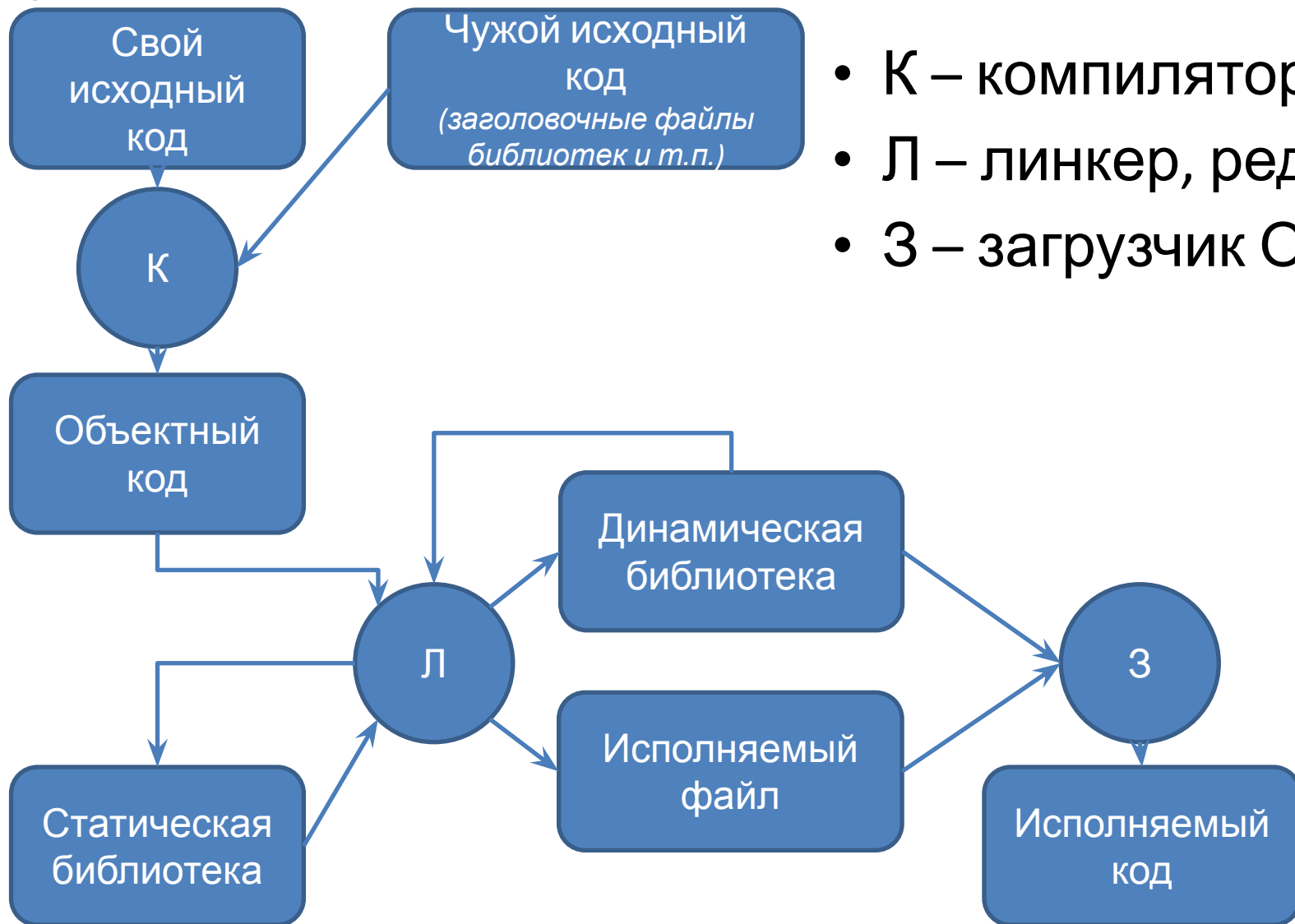
Сборка динамической библиотеки

- Вход: объектные файлы, статические библиотеки, ранее созданные динамические библиотеки
- Выход: файл со служебной информацией для ОС и машинными инструкциями, годными для исполнения процессором
- Динамическая библиотека – средство построения программ в процессе их работы

Сборка исполняемого файла

- Вход: объектные файлы, статические библиотеки, динамические библиотеки
- Выход: файл со служебной информацией для ОС, машинными инструкциями, годными для исполнения процессором, и «точкой входа»

Компиляция, сборка, загрузка в память для исполнения



- К – компилятор
- Л – линкер, редактор связей
- З – загрузчик ОС

Компиляция 1/3

worker.h

```
void do_some_work();
```

worker.c

```
#include "worker.h" // для чего нужна эта строка??  
void do_some_work() { /* ... */ }
```

main.c

```
#include "worker.h"  
int main()  
{  
    do_some_work();  
    return 0;  
}
```

КОМПИЛЯЦИЯ 2/3

```
c:\Users\espetrov>cl -c worker.c
```

```
c:\Users\espetrov>dumpbin /all /disasm worker.obj
```

<pre>Microsoft (R) COFF/PE Dumper Version 9.00.21022.08 Copyright (C) Microsoft Corporation. All rights reserved. Dump of file worker.obj File Type: COFF OBJECT FILE HEADER VALUES 14C machine (x86) 3 number of sections 5048208D time date stamp Thu Sep 06 11:03:25 2012 130 file pointer to symbol table 9 number of symbols 0 size of optional header 0 characteristics SECTION HEADER #1 .drectve name 0 physical address 0 virtual address 2F size of raw data 8C file pointer to raw data (0000008C to 000000BA) 0 file pointer to relocation table 0 file pointer to line numbers 0 number of relocations 0 number of line numbers 100A00 flags Info Remove 1 byte align RAW DATA #1 00000000: 20 20 20 2F 44 45 46 41 55 4C 54 4C 49 42 3A 22 /DEFAULTLIB:" 00000010: 4C 49 42 43 4D 54 22 20 2F 44 45 46 41 55 4C 54 LIBCMT" /DEFAULT 00000020: 4C 49 42 3A 22 4F 4C 44 4E 41 4D 45 53 22 20 LIB:"OLDNAMES" Linker Directives ----- /DEFAULTLIB:"LIBCMT" /DEFAULTLIB:"OLDNAMES"</pre>	<pre>SECTION HEADER #2 .debug\$\$ name 0 physical address 0 virtual address 70 size of raw data BB file pointer to raw data (000000BB to 0000012A) 0 file pointer to relocation table 0 file pointer to line numbers 0 number of relocations 0 number of line numbers 42100040 flags Initialized Data Discardable 1 byte align Read Only RAW DATA #2 00000000: 04 00 00 00 F1 00 00 00 61 00 00 00 23 00 01 11ñ...a...#... 00000010: 00 00 00 00 63 3A 5C 55 73 65 72 73 5C 65 73 70c:\Users\esp 00000020: 65 74 72 6F 76 5C 77 6F 72 6B 65 72 2E 6F 62 6A etrov\worker.obj 00000030: 00 3A 00 3C 11 00 22 00 00 07 00 0F 00 00 00 1E .:<.: "..... 00000040: 52 08 00 0F 00 00 00 1E 52 08 00 4D 69 63 72 6F R.....R..Micro 00000050: 73 6F 66 74 20 28 52 29 20 4F 70 74 69 6D 69 7A soft (R) Optimiz 00000060: 69 6E 67 20 43 6F 6D 70 69 6C 65 72 00 00 00 00 ing Compiler.... SECTION HEADER #3 .text name 0 physical address 0 virtual address 5 size of raw data 12B file pointer to raw data (0000012B to 0000012F) 0 file pointer to relocation table 0 file pointer to line numbers 0 number of relocations 0 number of line numbers 60500020 flags</pre>	<pre>Code 16 byte align Execute Read _do_some_work: 00000000: 55 push ebp 00000001: 8B EC mov ebp,esp 00000003: 5D pop ebp 00000004: C3 ret RAW DATA #3 00000000: 55 8B EC 5D C3 U.i]Ã COFF SYMBOL TABLE 000 0083521E ABS notype Static @comp.id 001 00000001 ABS notype Static @feat.00 002 00000000 SECT1 notype Static .drectve Section length 2F, #relocs 0, #linenums 0, checksum 0 004 00000000 SECT2 notype Static .debug\$\$ Section length 70, #relocs 0, #linenums 0, checksum 0 006 00000000 SECT3 notype Static .text Section length 5, #relocs 0, #linenums 0, checksum 672BE856 008 00000000 SECT3 notype () External _do_some_work String Table Size = 0x12 bytes Summary 70 .debug\$\$ 2F .drectve 5 .text</pre>
---	--	--

КОМПИЛЯЦИЯ 3/3

```
c:\Users\espetrov>cl -c main.c
```

```
c:\Users\espetrov>dumpbin /all /disasm main.obj
```

<p>Microsoft (R) COFF/PE Dumper Version 9.00.21022.08 Copyright (C) Microsoft Corporation. All rights reserved.</p> <p>Dump of file main.obj</p> <p>File Type: COFF OBJECT</p> <p>FILE HEADER VALUES</p> <p>14C machine (x86) 3 number of sections 50482092 time date stamp Thu Sep 06 11:03:30 2012 13D file pointer to symbol table A number of symbols 0 size of optional header 0 characteristics</p> <p>SECTION HEADER #1</p> <p>.drectve name</p> <p>0 physical address 0 virtual address 2F size of raw data 8C file pointer to raw data (0000008C to 000000BA) 0 file pointer to relocation table 0 file pointer to line numbers 0 number of relocations 0 number of line numbers</p> <p>100A00 flags</p> <p>Info Remove 1 byte align</p> <p>RAW DATA #1</p> <p>00000000: 20 20 20 2F 44 45 46 41 55 4C 54 4C 49 42 3A 22 /DEFAULTLIB:" 00000010: 4C 49 42 43 4D 54 22 20 2F 44 45 46 41 55 4C 54 LIBCMT" /DEFAULT 54 LIBCMT" /DEFAULT 00000020: 4C 49 42 3A 22 4F 4C 44 4E 41 4D 45 53 22 20 LIB:"OLDNAMES" Linker Directives ----- /DEFAULTLIB:"LIBCMT" /DEFAULTLIB:"OLDNAMES"</p>	<p>SECTION HEADER #2</p> <p>.debug\$\$ name</p> <p>0 physical address 0 virtual address 6C size of raw data BB file pointer to raw data (000000BB to 00000126) 0 file pointer to relocation table 0 file pointer to line numbers 0 number of relocations 0 number of line numbers</p> <p>42100040 flags</p> <p>Initialized Data Discardable 1 byte align Read Only</p> <p>RAW DATA #2</p> <p>00000000: 04 00 00 00 F1 00 00 00 5F 00 00 00 21 00 01 11f.....!... 00000010: 00 00 00 00 63 3A 5C 55 73 65 72 73 5C 65 73 70c:\Users\esp 00000020: 65 74 72 6F 76 5C 6D 61 69 6E 2E 6F 62 6A 00 3A etrov\main.obj.: 00000030: 00 3C 11 00 22 00 00 07 00 0F 00 00 00 1E 52 08 <.:.....R. 00000040: 00 0F 00 00 00 1E 52 08 00 4D 69 63 72 6F 73 6FR..Microso 00000050: 66 74 20 28 52 29 20 4F 70 74 69 6D 69 7A 69 6E ft (R) Optimizin 00000060: 67 20 43 6F 6D 70 69 6C 65 72 00 00 67 20 43 6F 6D 70 69 6C 65 72 00 00 g Compiler..</p> <p>SECTION HEADER #3</p> <p>.text name</p> <p>0 physical address 0 virtual address C size of raw data 127 file pointer to raw data (00000127 to 00000132) 133 file pointer to relocation table 0 file pointer to line numbers 1 number of relocations 0 number of line numbers</p> <p>60500020 flags</p> <p>Code 16 byte align Execute Read</p>	<p>_main:</p> <p>00000000: 55 push ebp 00000001: 8B EC mov ebp,esp 00000003: E8 00 00 00 00 call _do_some_work</p> <p>00000008: 33 C0 xor eax,eax 0000000A: 5D pop ebp 0000000B: C3 ret</p> <p>RAW DATA #3</p> <p>00000000: 55 8B EC E8 00 00 00 00 33 C0 5D C3 U.iè....3À]Ã</p> <p>RELOCATIONS #3</p> <table border="1"><thead><tr><th>Offset</th><th>Type</th><th>Symbol Applied To</th><th>Symbol Index</th><th>Symbol Name</th></tr></thead><tbody><tr><td>00000004</td><td>REL32</td><td></td><td>00000000</td><td>9 _do_some_work</td></tr></tbody></table> <p>COFF SYMBOL TABLE</p> <table border="1"><thead><tr><th>Address</th><th>Symbol</th><th>Section</th><th>Type</th><th>Attributes</th><th>Other</th></tr></thead><tbody><tr><td>000 0083521E</td><td>ABS</td><td></td><td>notype</td><td>Static</td><td> @comp.id</td></tr><tr><td>001 00000001</td><td>ABS</td><td></td><td>notype</td><td>Static</td><td> @feat.00</td></tr><tr><td>002 00000000</td><td>SECT1</td><td></td><td>notype</td><td>Static</td><td> .drectve</td></tr><tr><td></td><td></td><td>Section length</td><td>2F, #relocs</td><td>0, #linenums</td><td>0, checksum</td><td>0</td></tr><tr><td>004 00000000</td><td>SECT2</td><td></td><td>notype</td><td>Static</td><td> .debug\$\$</td></tr><tr><td></td><td></td><td>Section length</td><td>6C, #relocs</td><td>0, #linenums</td><td>0, checksum</td><td>0</td></tr><tr><td>006 00000000</td><td>SECT3</td><td></td><td>notype</td><td>Static</td><td> .text</td></tr><tr><td></td><td></td><td>Section length</td><td>C, #relocs</td><td>1, #linenums</td><td>0, checksum</td><td>226120D7</td></tr><tr><td>008 00000000</td><td>SECT3</td><td></td><td>notype</td><td>()</td><td>External</td><td> _main</td></tr><tr><td>009 00000000</td><td>UNDEF</td><td></td><td>notype</td><td>()</td><td>External</td><td> _do_some_work</td></tr></tbody></table> <p>String Table Size = 0x12 bytes</p> <p>Summary</p> <p>6C .debug\$\$ 2F .drectve C .text</p>	Offset	Type	Symbol Applied To	Symbol Index	Symbol Name	00000004	REL32		00000000	9 _do_some_work	Address	Symbol	Section	Type	Attributes	Other	000 0083521E	ABS		notype	Static	@comp.id	001 00000001	ABS		notype	Static	@feat.00	002 00000000	SECT1		notype	Static	.drectve			Section length	2F, #relocs	0, #linenums	0, checksum	0	004 00000000	SECT2		notype	Static	.debug\$\$			Section length	6C, #relocs	0, #linenums	0, checksum	0	006 00000000	SECT3		notype	Static	.text			Section length	C, #relocs	1, #linenums	0, checksum	226120D7	008 00000000	SECT3		notype	()	External	_main	009 00000000	UNDEF		notype	()	External	_do_some_work
Offset	Type	Symbol Applied To	Symbol Index	Symbol Name																																																																															
00000004	REL32		00000000	9 _do_some_work																																																																															
Address	Symbol	Section	Type	Attributes	Other																																																																														
000 0083521E	ABS		notype	Static	@comp.id																																																																														
001 00000001	ABS		notype	Static	@feat.00																																																																														
002 00000000	SECT1		notype	Static	.drectve																																																																														
		Section length	2F, #relocs	0, #linenums	0, checksum	0																																																																													
004 00000000	SECT2		notype	Static	.debug\$\$																																																																														
		Section length	6C, #relocs	0, #linenums	0, checksum	0																																																																													
006 00000000	SECT3		notype	Static	.text																																																																														
		Section length	C, #relocs	1, #linenums	0, checksum	226120D7																																																																													
008 00000000	SECT3		notype	()	External	_main																																																																													
009 00000000	UNDEF		notype	()	External	_do_some_work																																																																													

Сборка (линковка)

```
c:\Users\espetrov>link main.obj worker.obj /nodefaultlib /entry:main -out:main.exe
```

```
c:\Users\espetrov>dumpbin/all /disasm main.exe
```

<p>Microsoft (R) COFF/PE Dumper Version 9.00.21022.08 Copyright (C) Microsoft Corporation. All rights reserved.</p> <p>Dump of file main.exe</p> <p>PE signature found</p> <p>File Type: EXECUTABLE IMAGE</p> <p>FILE HEADER VALUES</p> <ul style="list-style-type: none">14C machine (x86)1 number of sections504828F7 time date stamp Thu Sep 06 11:39:19 20120 file pointer to symbol table0 number of symbolsE0 size of optional header103 characteristics<ul style="list-style-type: none">Relocations strippedExecutable32 bit word machine <p>OPTIONAL HEADER VALUES</p> <ul style="list-style-type: none">10B magic # (PE32)9.00 linker version200 size of code<ul style="list-style-type: none">0 size of initialized data0 size of uninitialized data1000 entry point (00401000)1000 base of code2000 base of data400000 image base (00400000 to 00401FFF)1000 section alignment200 file alignment5.00 operating system version0.00 image version5.00 subsystem version<ul style="list-style-type: none">0 Win32 version2000 size of image200 size of headers0 checksum3 subsystem (Windows CUI)8400 DLL characteristics<ul style="list-style-type: none">No structured exception handlerTerminal Server Aware	<ul style="list-style-type: none">100000 size of stack reserve1000 size of stack commit100000 size of heap reserve1000 size of heap commit0 loader flags10 number of directories<ul style="list-style-type: none">0 [0] RVA [size] of Export Directory0 [0] RVA [size] of Import Directory0 [0] RVA [size] of Resource Directory0 [0] RVA [size] of Exception Directory0 [0] RVA [size] of Certificates Directory0 [0] RVA [size] of Base Relocation Directory0 [0] RVA [size] of Debug Directory0 [0] RVA [size] of Architecture Directory0 [0] RVA [size] of Global Pointer Directory0 [0] RVA [size] of Thread Storage Directory0 [0] RVA [size] of Load Configuration Directory0 [0] RVA [size] of Bound Import Directory0 [0] RVA [size] of Import Address TableDirectory<ul style="list-style-type: none">0 [0] RVA [size] of Delay Import Directory0 [0] RVA [size] of COM Descriptor Directory0 [0] RVA [size] of Reserved Directory	<p>SECTION HEADER #1</p> <ul style="list-style-type: none">.text name15 virtual size1000 virtual address (00401000 to 00401014)200 size of raw data200 file pointer to raw data (00000200 to 000003FF)0 file pointer to relocation table0 file pointer to line numbers0 number of relocations0 number of line numbers <p>60000020 flags</p> <ul style="list-style-type: none">CodeExecute Read <pre>00401000: 55 push ebp 00401001: 8B EC mov ebp,esp 00401003: E8 08 00 00 00 call 00401010 00401008: 33 C0 xor eax,eax 0040100A: 5D pop ebp 0040100B: C3 ret 0040100C: CC int 3 0040100D: CC int 3 0040100E: CC int 3 0040100F: CC int 3 00401010: 55 push ebp 00401011: 8B EC mov ebp,esp 00401013: 5D pop ebp 00401014: C3 ret</pre> <p>RAW DATA #1</p> <pre>00401000: 55 8B EC E8 08 00 00 33 C0 5D C3 CC CC CC U.iè....3À]Ãìììì 00401010: 55 8B EC 5D C3 U.ijÃ</pre> <p>Summary</p> <ul style="list-style-type: none">1000 .text
--	--	--

Отладка

- Достижение работоспособности программы, устранение грубых ошибок
- Методы отладки
 - Имитация пошагового исполнения с помощью «карандаша и бумаги» для простых случаев
 - Трассировка работы программы с помощью отладочной печати
 - Проверка необходимых условий корректности в ходе работы программы
 - Пошаговое исполнение программы с помощью отладчика

Оптимизация

- Улучшение количественных характеристик программы
 - Время компиляции
 - Время загрузки
 - Время работы
 - Размер используемой памяти (данных на диске)
 - Размер исходного кода
 - Размер исполняемого кода
- Компилятор и линкер умеют автоматически делать *некоторые* преобразования программ
 - Сохраняют корректность программы
 - Могут менять некорректную программу неожиданным образом
 - Могут *ухудшать* количественные характеристики программы

Заключение

- Информация об этом курсе
- Понятие программы
- Этапы разработки программ
 - Написание исходного кода
 - Компиляция исходного кода
 - Сборка
 - Отладка
 - Оптимизация
 - Тестирование