

Соловьев С.Ю.

**Дополнительные главы
программной инженерии**

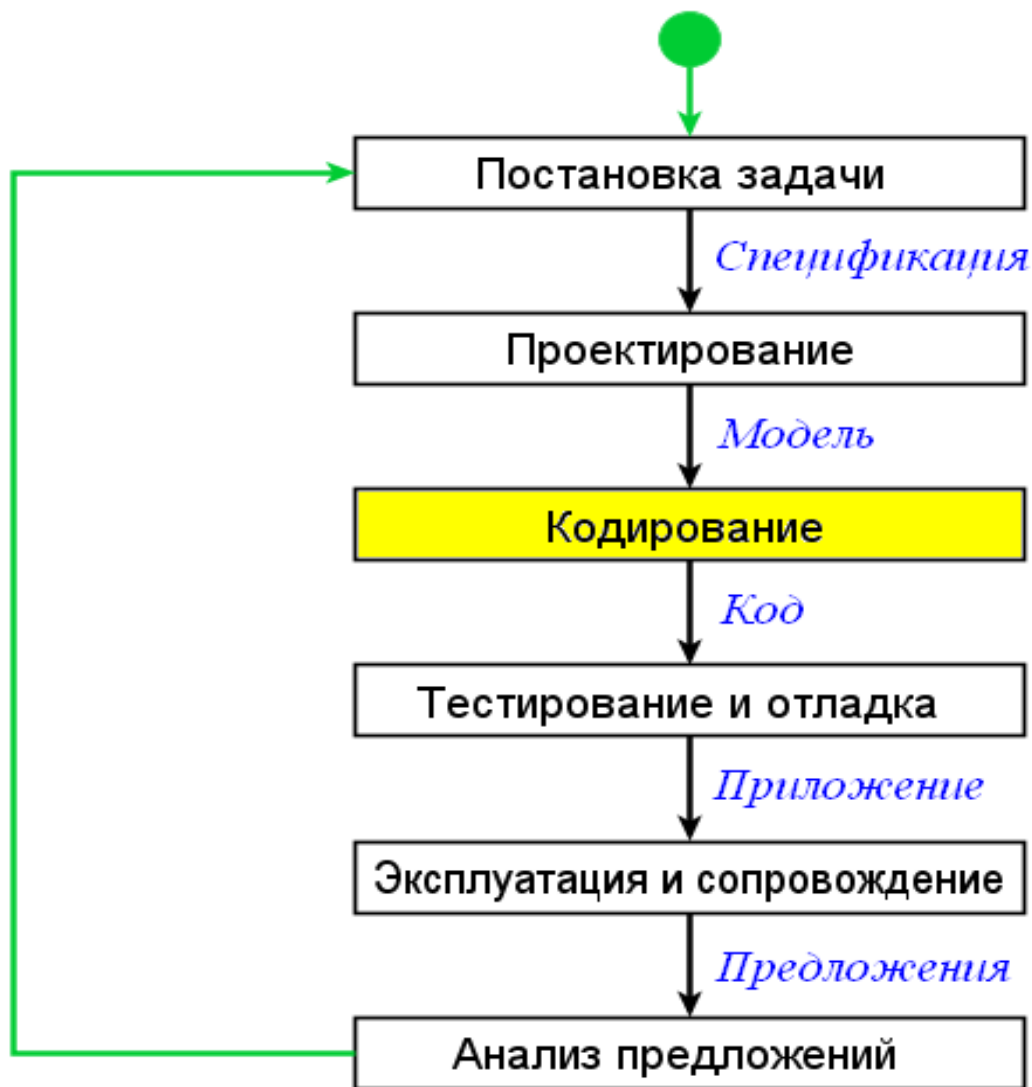
(7)

Кодирование

2021

Напоминание

Жизненный цикл



Кодирование

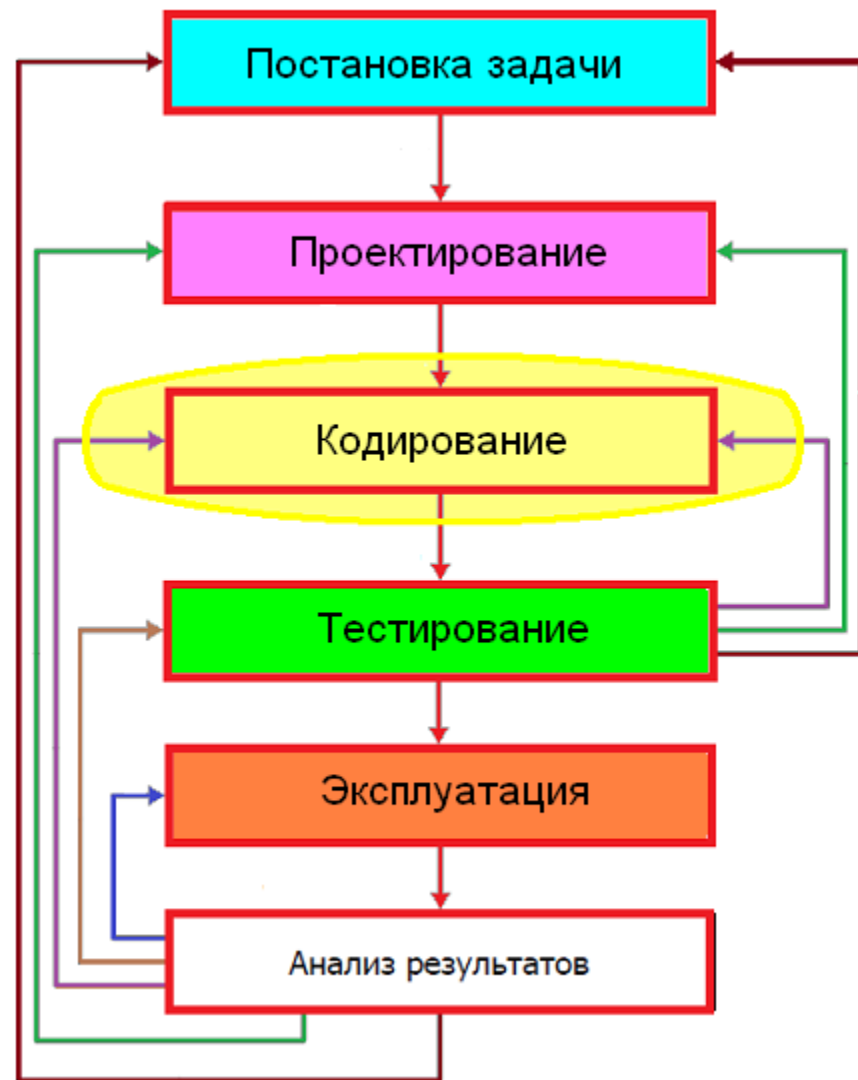
= Программирование =
Реализация

- коллективный труд ИТ-специалистов
по конструированию цифровых продуктов
в соответствии с планом разработки, (1)
детальным проектом, (2)
архитектурой и (3)
требованиями. (4)

конструирование

- Программирование
- Подготовка информационного наполнения
- Техническое документирование

План разработки ПО (Затраты)

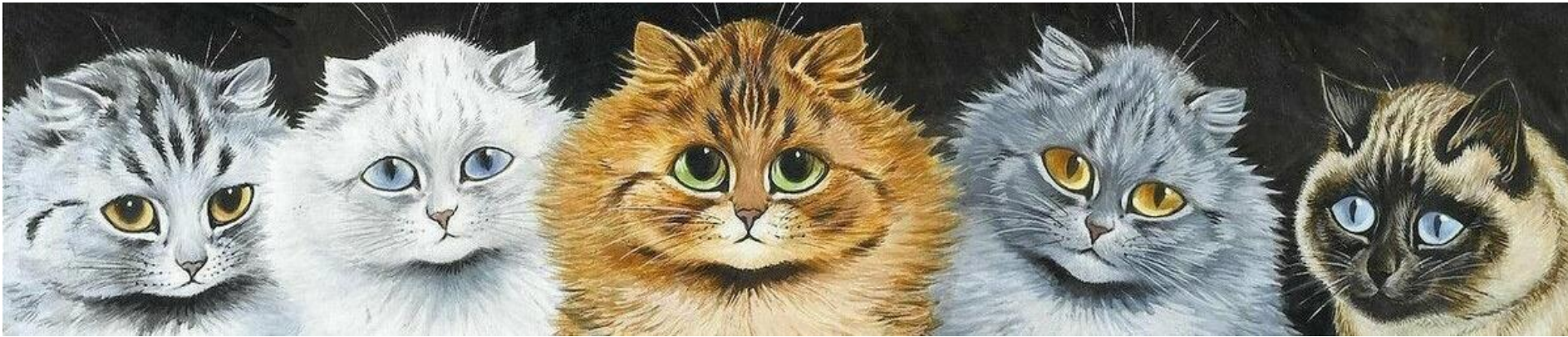


План разработки ПО

График выполнения работ по разработке Программно-информационного комплекса «Планирование развития объектов теплоснабжения города Москвы»

ЭТАПЫ РАБОТ	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль	Март	Апрель
Разработка частного технического задания на ПИК «ПР»														
Создание информационно-графического описания объектов тепловых сетей в цифровой расчетно-аналитической модели														
Описание топологической связности объектов системы теплоснабжения														
Внедрение, <u>разработка</u> или/и модернизация модулей				Разработка										
Отладка, калибровка расчетных модулей ПИК «ПР»														
Ввод в опытную эксплуатацию														
Разработка модуля обмена данными с информационными системами ОАО «МОЭК»														

Технологии коллективной разработки



- Авторская разработка 
- Коллективная разработка

[Lovis Wain]

- равноправные исполнители;
- бригада главного программиста;
- программирование в парах;
- ядерная модель;
- проектная группа MSF;

программист,
тестировщик,
администратор БД,
системный администратор,
системный интегратор,
технический писатель, ...

- Общинная модель разработки
- Офшорное программирование

[Рейнвотер]



IDE: Integrated
Interactive

development environment

Интегрированная среда разработки – интерактивная оболочка, объединяющая инструментальные средства разработки цифровых продуктов.



1983 Turbo

1995 Visual
Studio

--//-- Delphi

1997 NetBeans

2003 Xcode

IDE-проект

IDE-проект – предмет труда, на который направлена разработка цифрового продукта.

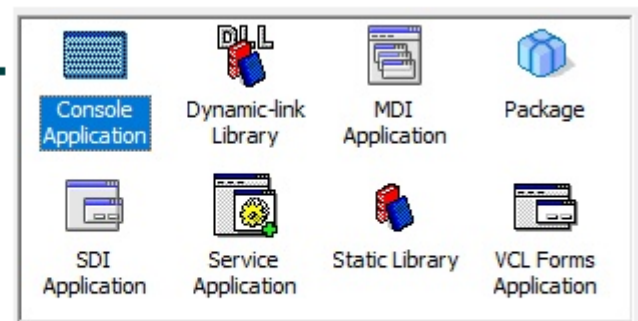
проект

Файл проекта
Файл параметров проекта
Файлы исходных модулей
Файлы объектных модулей
Файлы библиотек
Исполняемый файл
Файлы форм
Файлы ресурсов
Файлы изображений
Файлы справки
Файлы резервных копий
...

Управление проектом

Проект: **создать** ►
удалить
открыть
закрыть
копировать
экспортировать
импортировать

Add...
Remove File...
Save
Options...
Activate
Compile
Build
View Source



Редактор [PHP; netbeans.org]

1/5

Выделение синтаксиса ❶

Свертывание кода ❷

Навигатор по структуре кода ❸

Автозавершение выражений и кода ❹

Переход к объявлению из позиции использования

Генерация кода: конструкторов, методов получения/установки и др.

Форматирование фрагмента кода

Сообщения об ошибках ❺

Подсказка обращений к функциям

```
14 class Bug
15 {
16     private $products = null;
17
18     public function assignToProduct($product)
```

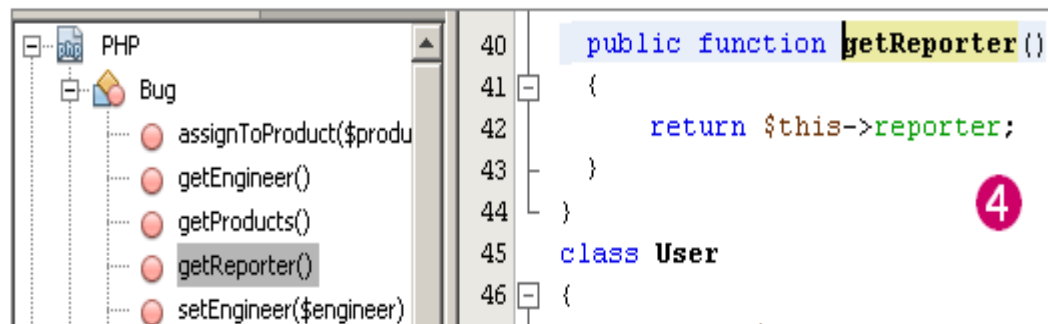
❶

```
45 class User
46 {
47     private $assignedBugs = null;
48     public function assignedToBug($bug)
49     {
50         $this->assignedBugs[] = $bug;
51     }
52 }
```

❷

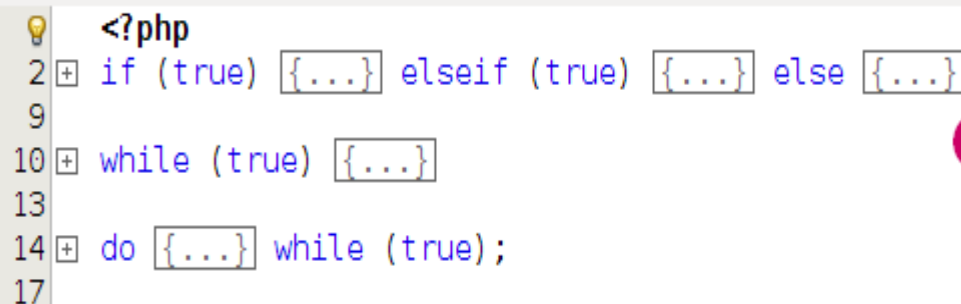
```
51 public function addReportedBug($bug)
{
    $this->reportedBugs[] = $bug;
```

❸



```
40 public function getReporter()
41 {
42     return $this->reporter;
43 }
44 }
45 class User
46 {
```

❹



```
<?php
2 if (true) {...} elseif (true) {...} else {...}
9
10 while (true) {...}
13
14 do {...} while (true);
17
```

❺

Редактор

Правила кодирования

Стиль оформления исходного кода

Структурирование	Разрежение	Именование	
1 оператор – 1 строка	пробелы	переменных	функций
отступы	пустые строки		
операторные скобки	Стиль оформления исходного кода		

Стиль программирования

Защитное программирование	Устойчивый интерфейс		
	Контроль	пред- и постусловий	
		ввода/вывода	
		индексов и указателей	
		арифметических операций	

Имя функции начинается глаголом



Выравнивание



```
function MakeWindow
  (var W : WindowPtr;
   XLow, YLow : Byte;
   XHigh, YHigh : Byte;
   DrawFrame : Boolean;
   ClearWindow : Boolean;
   SaveWindow : Boolean;
   WindowAttr : Byte;
   FrameAttr : Byte;
   HeaderAttr : Byte;
   Header : string
  ) : Boolean;
  {-Allocate and initialize, but do not display, a new window}
```

Назначение
функции

Осмысленные
имена параметров



название этапа



пробел после запятой



Защитное
программирование



отступ

```
{Write the header}
BlockWrite(F, AP^.Header, SizeOf(ArrayHeader), Bytes);
if (IoResult <> 0) or
  (Bytes <> SizeOf(ArrayHeader)) then begin
  Close(F);
  P := IoResult;
  Error(9);
  Exit;
end;
```

стиль Кернигана и Ричи

Система стандартов и методик разработки конфигураций

Соглашения при написании кода

для платформы 1С:Предприятие 8
(<https://its.1c.ru/db/v8std>)

Оформление модулей

Редактор

Правила образования имен переменных

Область применения: управляемое приложение, мобильное приложение, обычное приложение.

1. Имена переменных следует образовывать от терминов предметной области таким образом, чтобы из имени переменной было понятно ее назначение.

2. Имена следует образовывать путем удаления пробелов между словами. При этом, каждое слово в имени пишется с прописной буквы. Предлоги и местоимения из одной буквы также пишутся прописными буквами. Пример:

Перем ДиалогРаботыСКаталогом; // Диалог работы с каталогом
Перем КоличествоПачекВКоробке; // Количество пачек в коробке

Примеры некорректных имен переменных:

масРеквизитов, соотвВидИмя, новСтр

3. Имена переменных запрещается начинать с подчеркивания.

4. Имена переменных не должны состоять из одного символа. Использование односимвольных имен переменных допускается только для счетчиков циклов.

5. Переменные, отражающие состояние некоторого флага, следует называть так, как пишется истинное значение этого флага.

Стандарты именования

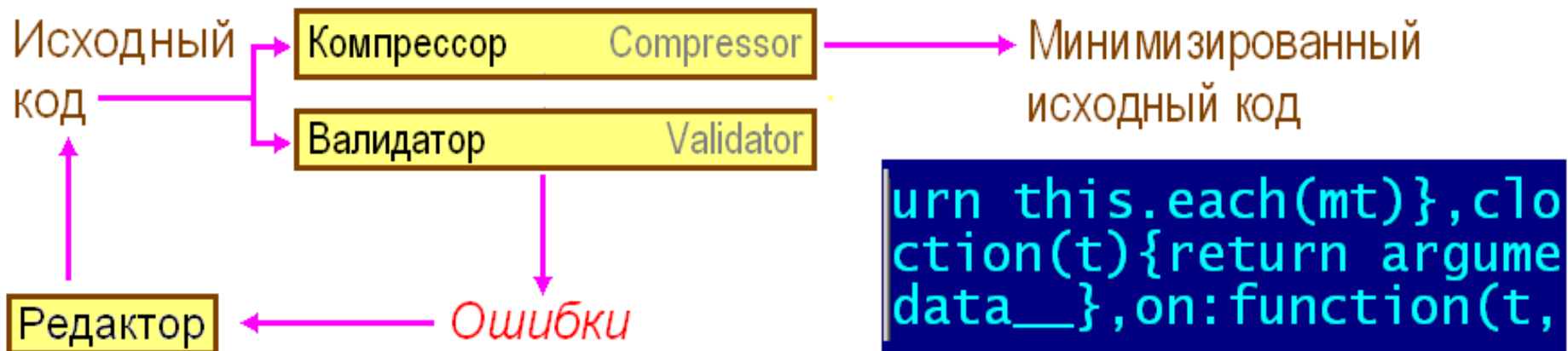
Префиксы венгерской нотации для 32-разрядных переменных [Финогенов]

Префикс	Расшифровка	Значение
b	Bool	Логическая (булева) переменная, 32 бита
c	Character	Символ, 1 байт
dw	DoubleWord	Двойное слово без знака, 32 бита
f	Function	Функция
h	Handle	Дескриптор объекта
l	long	Длинное целое со знаком, 32 бита
lp	LongPointer	Дальний указатель, 32 бита
lpz	LongPointer StringZero	Дальний указатель на символьную строку, заканчивающуюся двоичным нулем, 32 бита
n	iNt	Целое со знаком, 32 бита
p	Pointer	Ближний указатель, 32 бита
pt	PoinT	X- и y-координаты точки, упакованные в 64 бита
rgb	RedGreenBlue	Цвет из красной, зеленой и синей составляющих, упакованный в 32 бита
sz	StringZero	Символьная строка, заканчивающаяся двоичным нулем
u	UInt	Целое без знака, 32 бита
w	Word	Слово без знака, 16 бит

Транслятор



LISP, JavaScript, PHP, Perl



Библиотеки

Библиотека объектных модулей – набор объектных модулей, рассчитанных на применение в прикладных программах.

	Статические	Динамические
Windows	.lib	.dll
Linux	.a	.so

Файл DLL	Версия	Описание
 aaclient.dll	6.1.7601.18079	Клиент доступа из любого места
 acctres.dll	6.1.7600.16385	Ресурсы диспетчера учетных записей
 acledit.dll	6.1.7600.16385	Редактор списка доступа ACL
 aclui.dll	6.1.7600.16385	Редактор дескрипторов безопасности
 acppage.dll	6.1.7601.17514	Библиотека расширений для вкладки
 acproxy.dll	6.1.7600.16385	Библиотека DLL прокси AUTOCHK

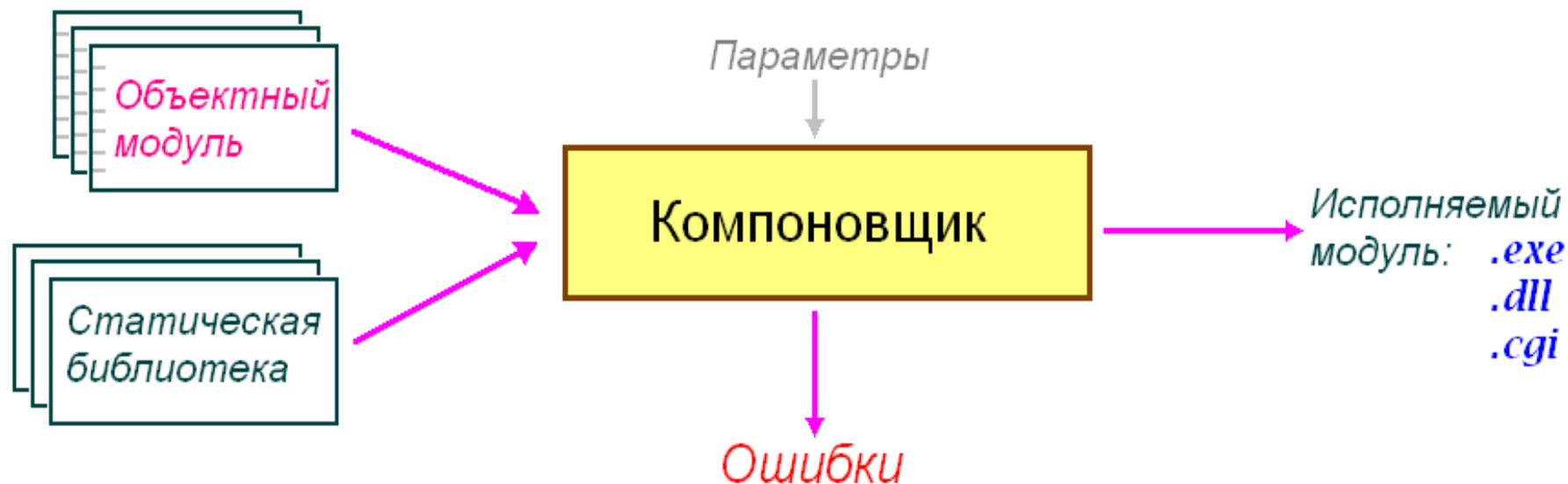
Библиотека исходных модулей – набор исходных модулей, рассчитанных на применение в цифровых продуктах.

D3.js – библиотека JavaScript для создания интерактивных визуализаций



Инструментарий сборки

1/3



Загрузить библиотеку

```
<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
```

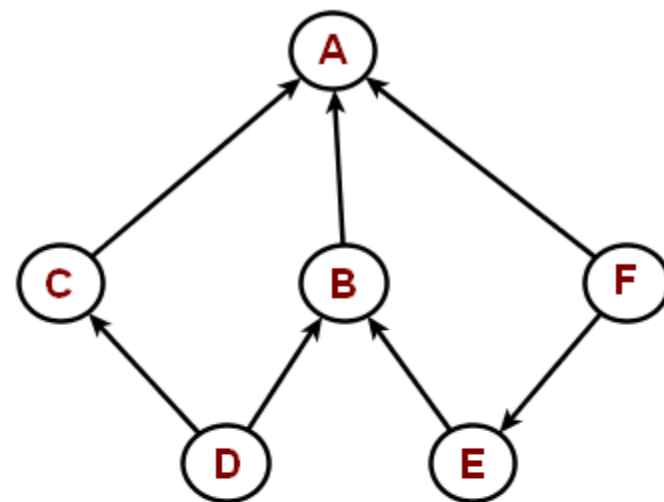
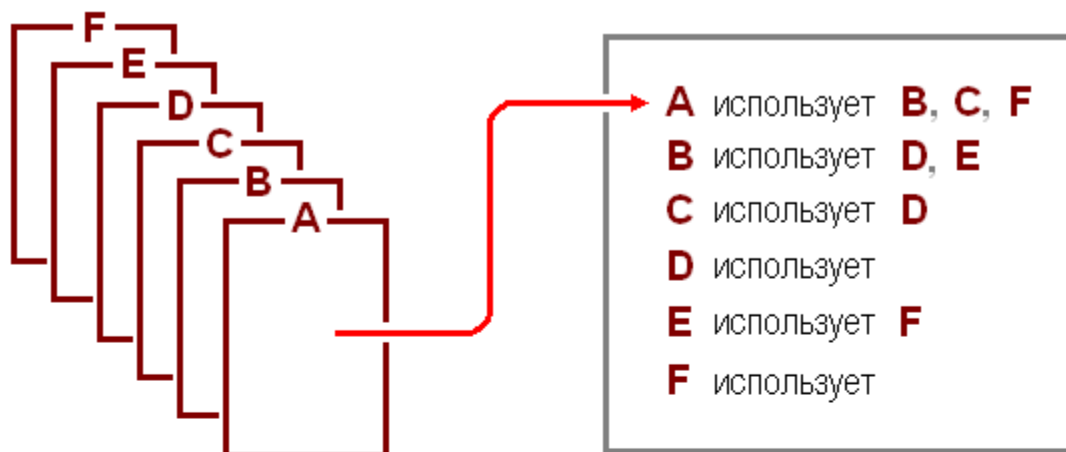


Создать графический эл-т

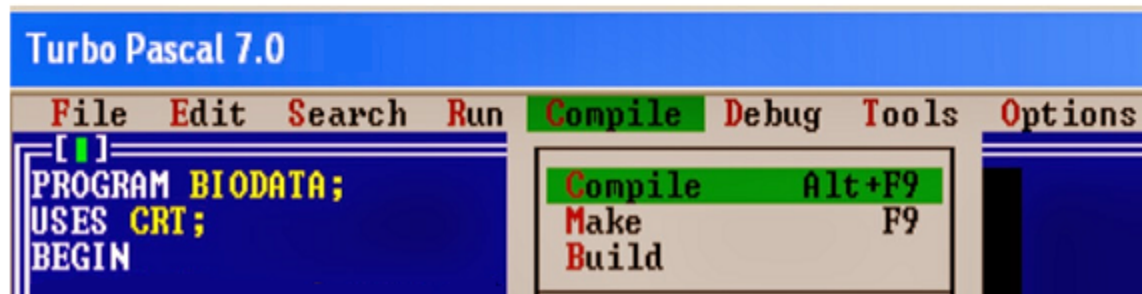
```
d3.select('body').append('div').classed('chart_area', true);
```

Инструментарий сборки

2/3



*Порядок
компиляции*



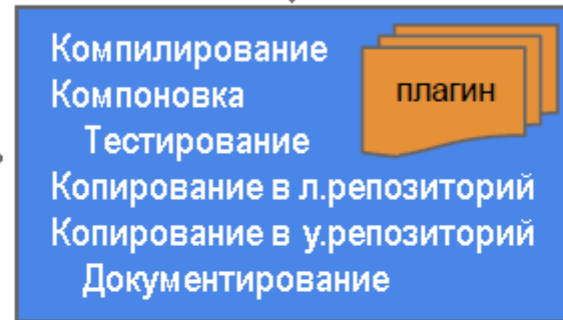
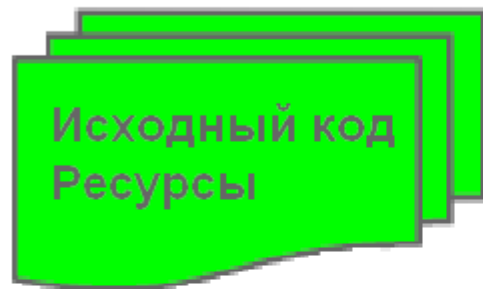
- I. **D, F**
- II. **C, E**
- III. **B**
- IV. **A**

Инструментарий сборки

3/3

(≈ Maven)

Описание сборки



Готовый продукт



Локальный репозиторий

Центральный репозиторий

Удаленный репозиторий





Отладчик / Символьный отладчик

Точки останова

Шаг с заходом

Шаг с обходом

Шаг с выходом

Условия останова

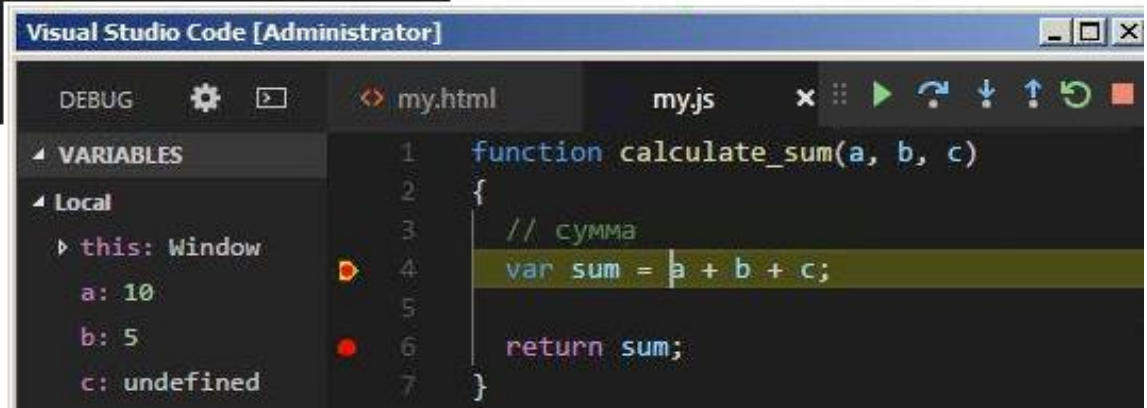
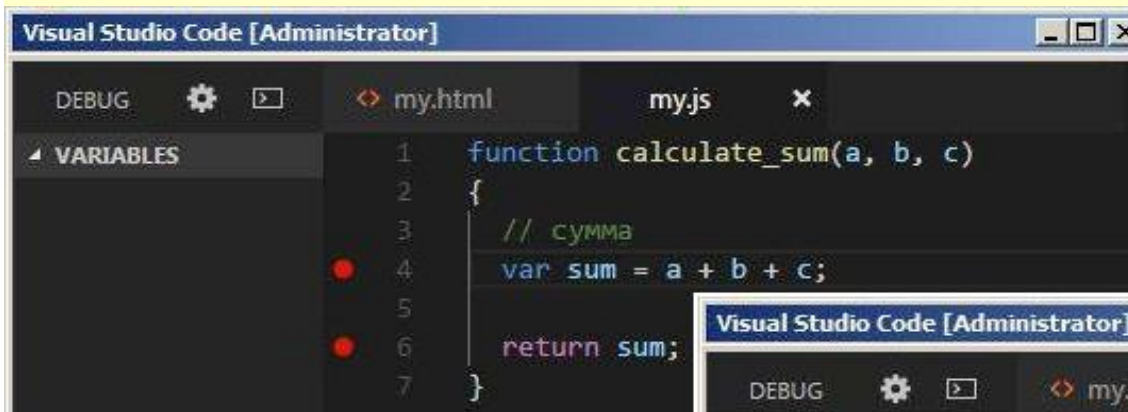
Отслеживание событий с занесением результатов в журнал

Выполнить до текущей позиции

Продолжить до конца

Подсказка обращений к функциям

Вычисления в текущем контексте





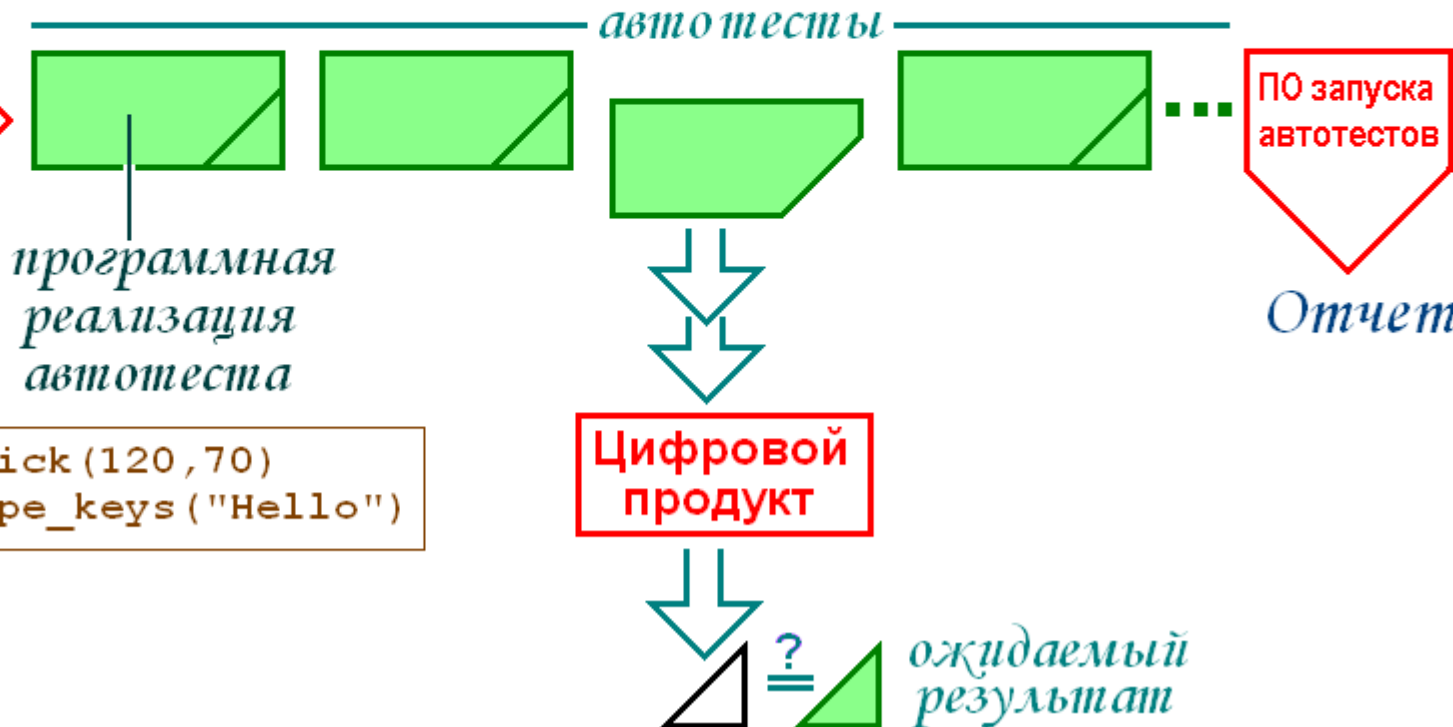
Инструментарий тестирования

1/3

Теория тестирования:



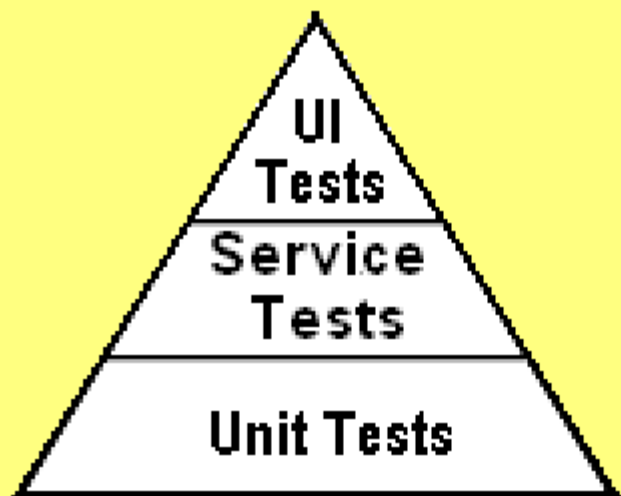
инструментарий
подготовки
автотестов



1. **Организатор тестов** управляет выполнением тестов.
2. **Генератор тестовых данных**
 - выбирает тестовые данные из БД или
 - генерирует их случайным образом по спец. шаблонам.
3. **Оракул** генерирует ожидаемые результаты тестов.
Оракулом могут быть предыдущие версии ПО. **N.B.**
4. **Компаратор файлов**
 - сравнивает результаты текущего тестирования
с результатами предыдущего тестирования и
 - составляет отчет о различиях.
5. **Генератор отчетов** о результатах тестирования.

Пирамида автоматизации тестов

(Кон М. SCRUM: Гибкая разработка ПО, 2010)



Пользовательское



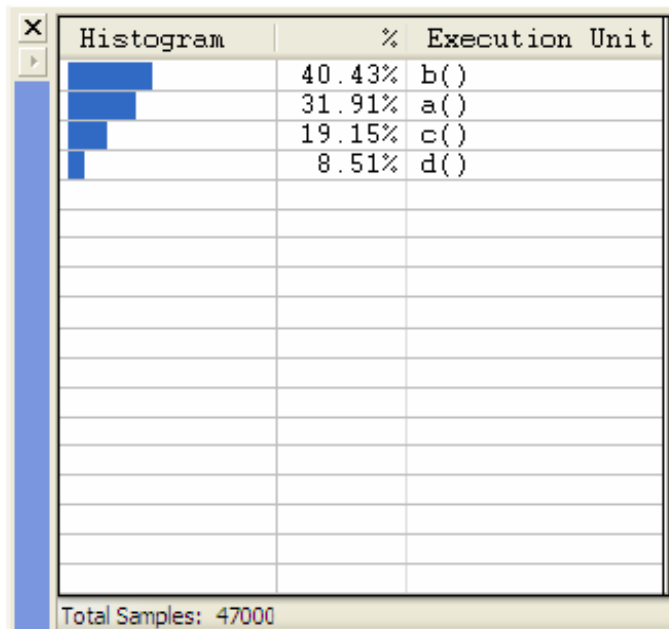
Сервисное

Модульное

Профилировщик (Profiler)

– программный инструмент, который **анализирует** программу во время ее работы (на тестовом примере) и **собирает** информацию

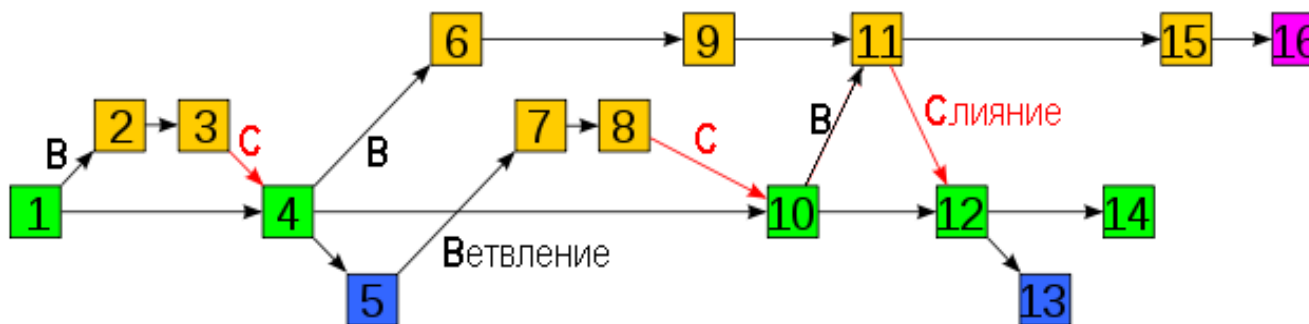
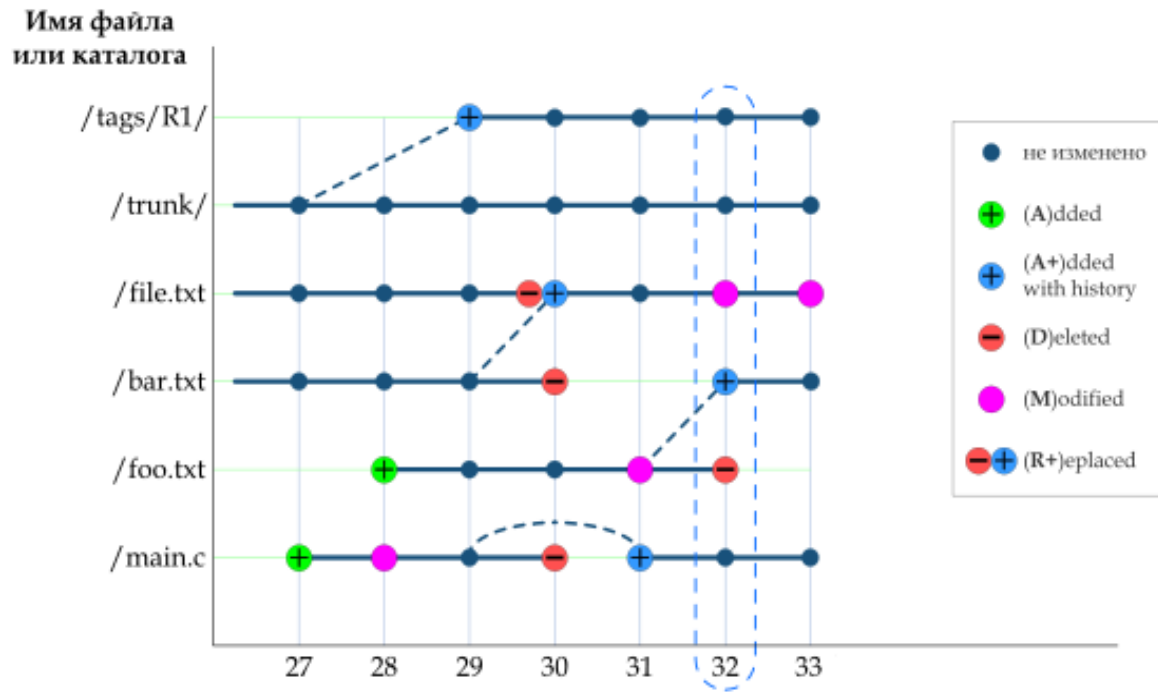
- временной сложности,
- выделенной памяти,
- критических участках и
- об использовании конкретных методов.



%	Line	Source
	27	void b()
	28	{
	29	int i;
	30	int array[MAX];
18.11%	31	for (i=0; i < MAX; i++)
	32	{
22.32%	33	array[i] = i*i;
	34	}
	35	}
	36	void c()
	37	{
	38	int i;
	39	int array[MAX/2];
8.60%	40	for (i=0; i < (MAX/2); i++)
	41	{
10.55%	42	array[i] = i*i
	43	}
	44	}

Управление версиями

(Version Control System; VCS)



Типовая ветвь

Ветви

Стебловая ветвь

Метки / Теги

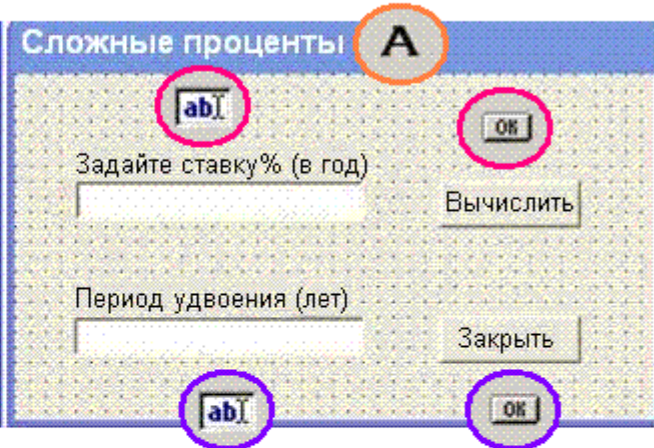
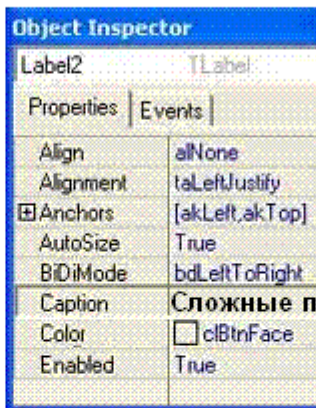
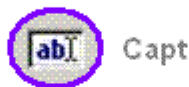
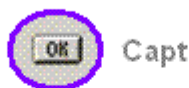
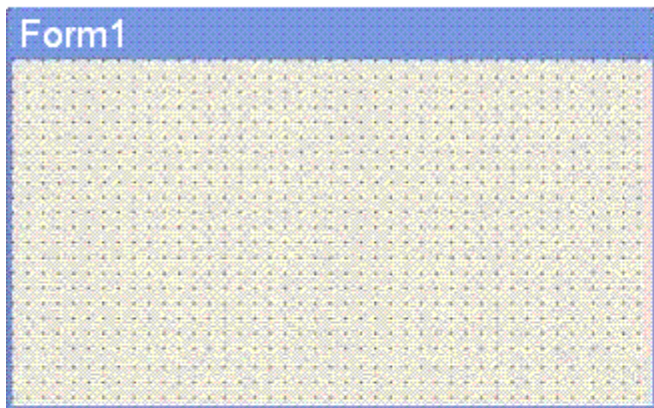
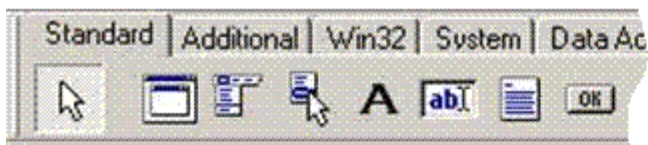




Конструктор пользовательских интерфейсов

(Визуальное программирование) [DELPHI]

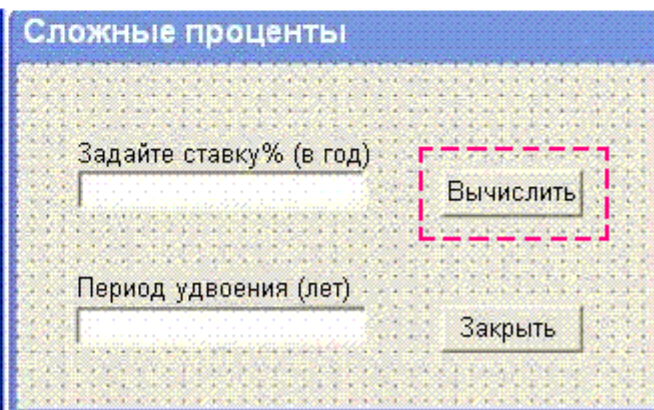
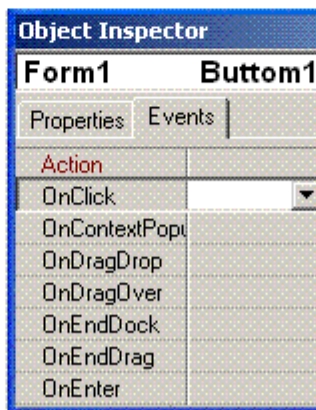
Палитра компонентов



```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
end;
```



```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Edit2.Text := IntToStr(72 div StrToInt(Edit1.Text));  
end;
```



Инструментарий справочной системы

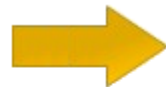
(MS Compressed HTML Help; .chm, 1997)

HELP



help
проект

Содержание
Страницы html
Файлы изображений



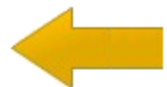
.chm



+

.dll

Содержание()
Поиск()
Указатель()
Избранное()



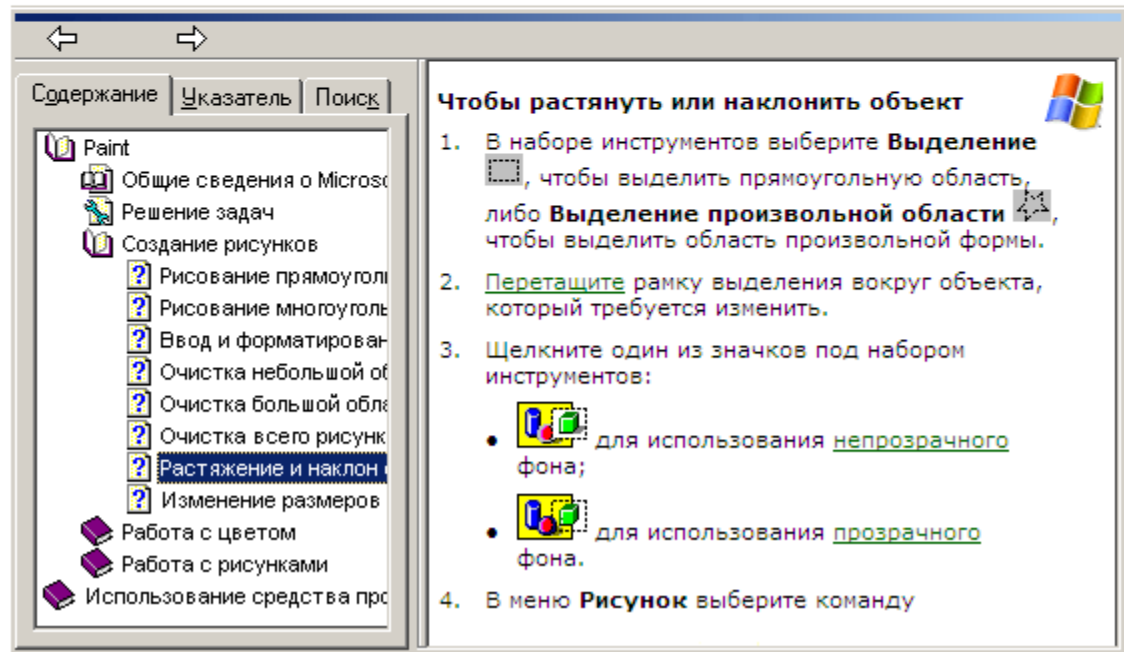
Заголовок

Ключевые слова

СПРАВКА

Гиперссылки

Якоря



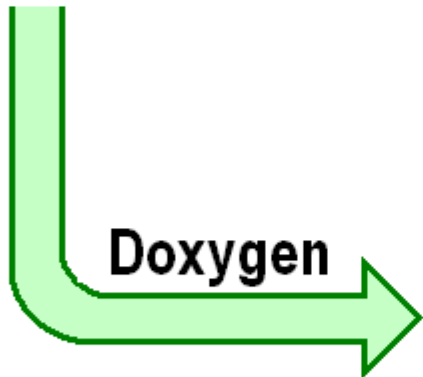
Генератор документации

Исходный код на C++

```
/*!  
Находит сумму двух слагаемых  
\param a,b Слагаемые  
\return Сумму слагаемых, переданных в качестве аргументов  
*/  
double sum(const double a, const double b);  
{ return a + b; }
```

Руководство
программиста

Фрагмент документации



Doxygen

Документирование элементов исходного
кода: файлов, классов, функций, методов.

Функции

```
double sum ( const double a,  
             const double b  
            )
```

Находит сумму двух слагаемых

Аргументы

a,b Слагаемые

Возвращает

Сумму слагаемых, переданных в качестве аргументов

Инструментарий рефакторинга

Рефакторинг – изменение исходного кода без изменения его внешнего поведения

Фаулер М.
Рефакторинг.
Улучшение
существующего
кода, 1999



Запахи кода Bad smells Плохие запахи

Дублирование кода
Длинный метод
Большой класс
Ленивый класс
Группы данных
Класс данных
Посредник

≈ 60 запахов

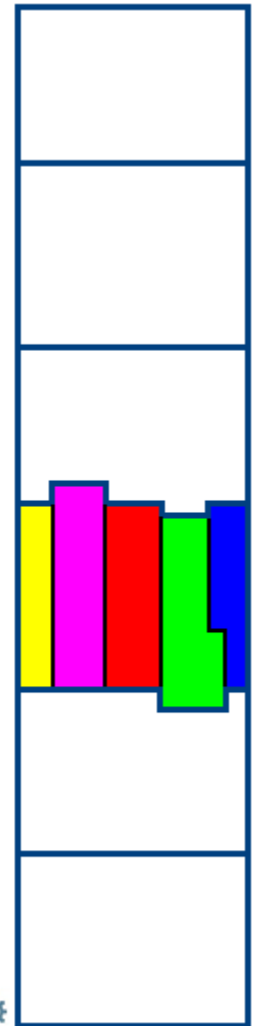


Анализ кода

Методики устранения
(рефакторинги)



переименования ⚙
перемещения ⚙
обобщения кода ⚙
технические генерации ⚙
изменения модели классов ⚙





Инструментарий доступа к БД

1/3

цель

База данных – удобно организованная *совокупность данных*, доступных для обработки приложениями заданного класса.

права доступа

СУБД

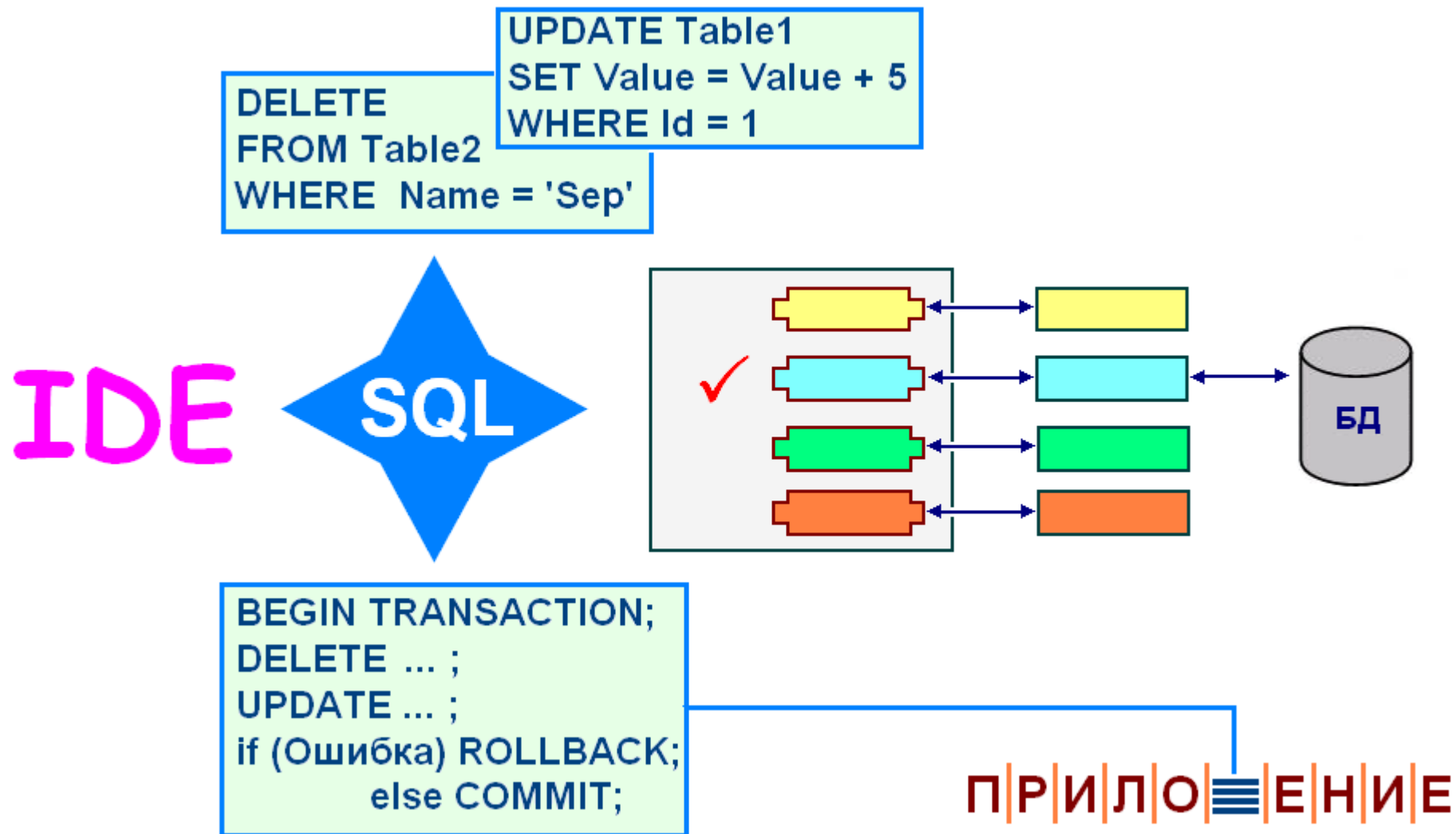


Жизненный цикл базы данных



Инструментарий доступа к БД

3/3



Анализ исходного кода

(аудит, инспекция)



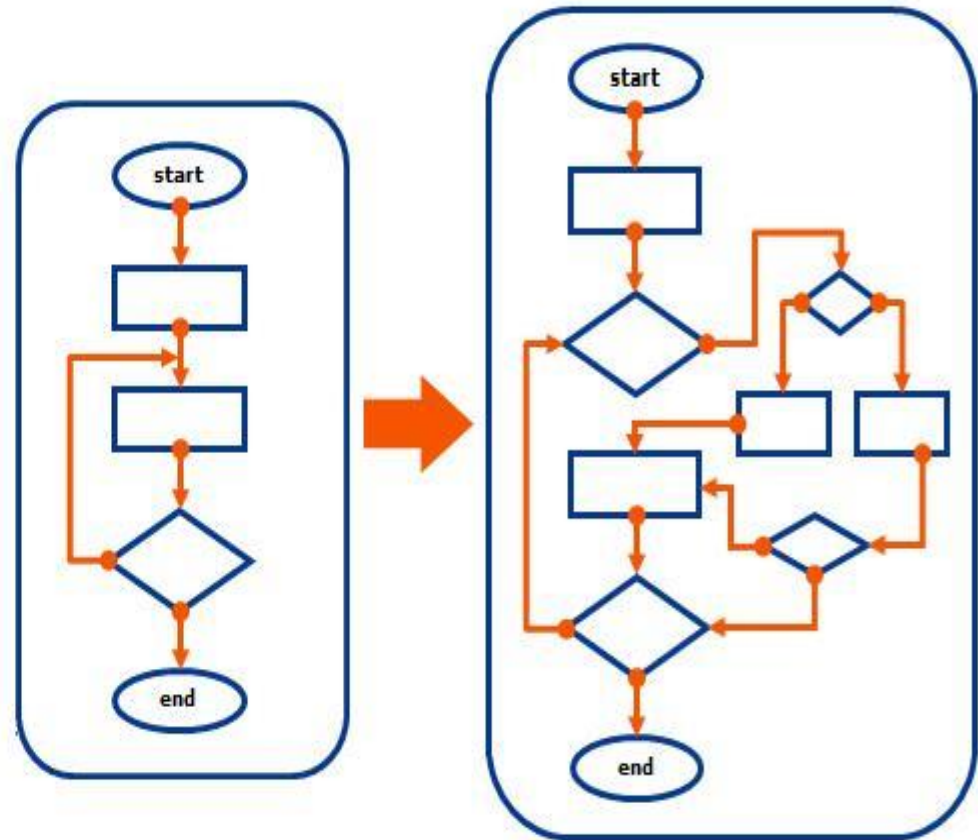
Запутывание кода (Обфускация)

1/2

– преобразование программного кода к виду сложному для изучения посторонними лицами.

- лексическая обфускация /5
- превентивная обфускация
- обфускация структур данных /3
- обфускация потока управления

- добавления недостижимого кода
- добавления "мертвого" кода
- добавления избыточного кода
- "переплетения" функций
- "клонирования" функций
- распараллеливания кода
- устранения библиотечных вызовов



Компрессия **vs.** Лексическая обфускация

2/2

```
int    COUNT = 100;
float  TAX_RATE = 0.2;

for (int i=0; i<COUNT; i++)
{
    tax[i] = orig_price[i] * TAX_RATE;
    price[i] = orig_price[i] + tax[i];
}
```

i → a
tax → b
org_price → c
price → d

Компрессия

```
for(int a=0;a<100;a++){b[a]=c[a]*0.2;d[a]=c[a]+b[a];}
```

Обфускация

TAX_RATE → jht4hnv
i → plf5ojvb
tax → weq83efas_23
org_price → RwB1AHIA
price → xX03123012

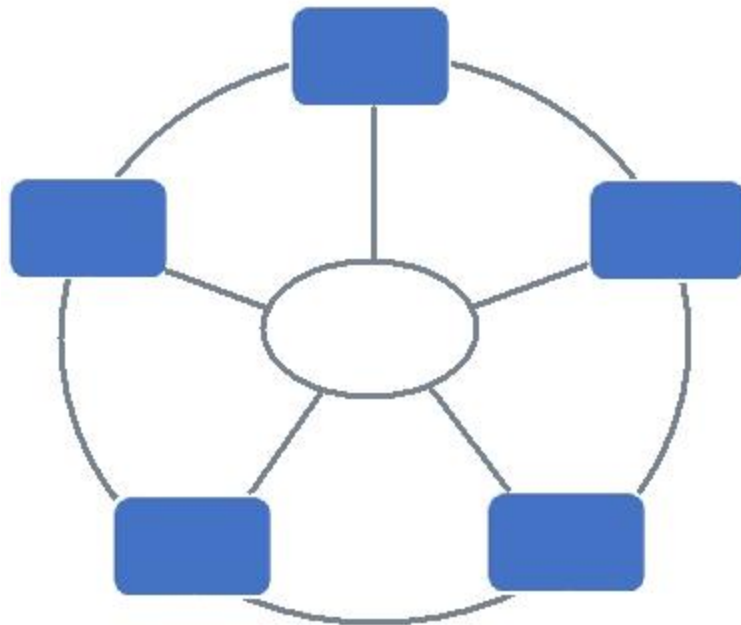
```
float jht4hnv=0.2;for(int plf5ojvb=0;
plf5ojvb<100;plf5ojvb++){weq83efas_23
[plf5ojvb]=RwB1AHIA[plf5ojvb]*jht4hnv;
xX03123012[plf5ojvb]=RwB1AHIA[plf5ojvb]
+weq83efas_23[plf5ojvb];}
```

Инструментарий командной разработки

1/2



Аналитик Менеджер Архитектор Разработчик Тестировщик



Союз равноправных



Жесткая иерархия

Инструментарий командной разработки

Azure DevOps Server бывш. Team Foundation Service платформа для коллективной разработки проектов

Рабочий элемент – отдельная единица (шаг) работы.

Типы элементов: Ошибка, Задача, Требование, Сценарий и т.д.

Состояние

Права доступа
История эл-та

Уведомления
Методики контроля
Допустимые действия

Диаграмма
переходов
состояний



Команда проекта

отслеживание проекта
наблюдение за рабочими элементами
просмотр отчетов
общение членов команды

В о п р о с ы?

soloviev@glossary.ru

Соловьев С.Ю.

Основы программной инженерии

www.park.glossary.ru