

Университет машиностроения
Кафедра «Автоматика и процессы управления»

Дисциплина
Информационные технологии
2 семестр

Тема 01

Программирование и
основы алгоритмизации
(с использованием VBA for Ms Excel)

Место раздела в структуре курса:

1. Устройства и принципы функционирования вычислительных машин.
2. Прикладное программное обеспечение.
3. **Основы алгоритмизации и программирования с включенной средой VBA (Visual Basic for Application).**
4. Технология (методы) разработки программного обеспечения.
5. Современные подходы к разработке информационных систем (объектно-ориентированное программирование).
- 5+ Управление техническими системами.

Основы алгоритмизации и программирования в среде VBA for Excel (Visual Basic for Application):

- 1. Средства автоматизации в Microsoft Office**
- 2. Среда разработки IDE VBA и редактирование макросов**
- 3. Базовые алгоритмические конструкции**
- 4. Взаимодействие с хост-приложением и элементы управления**
- 5. Разработка приложений на основе пользовательских форм и визуальных компонент**
- 6. Работа с текстовыми файлами**
- 7. Взаимодействие с системами управления базами данных**
- 8. Работа в распределённых вычислительных сетях**

Список рекомендуемой литературы:

1. Кузьменко В.Г., VBA 2002. Самоучитель. – 2-е изд. – М.: ООО «Бином-Пресс», 2004 г. – 416 с.: ил.
2. Гарбер Г.З., Основы программирования на Visual Basic и VBA в Excel 2007. – М.: СОЛОН-ПРЕСС, 2008. – 192 с.
3. Фрай К., Фриз В., Бакингом Ф., Программирование в Office Excel 2003. – С.Пб.: ИД «Пи-тер», 2004;
4. Гарнаев А.Ю., Использование Ms Excel и VBA в экономике и финансах. – С.Пб.: БХВ – Санкт-Петербург, 1999

ГОСТ

19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

Список рекомендуемых сайтов:

- 1. Википедия (<http://ru.wikipedia.org>);**
- 2. Викиучебник (<http://ru.wikibook.org>) раздел «Информационные технологии»;**
- 3. Электронная библиотека (<http://wmate.ru/ebooks/cat10/>);**
- 4. VBA в MSDN (<http://support.microsoft.com/kb/163435/ru>);**
- 5. База готовых кодов на Visual Basic (<http://www.vbfaq.ru/>);**
- 6. Справочник по VBA (<http://vba-help.ru/>);**
- 7. Литература по Excel и VBA (<http://excelvba.ru/books>)**
- 8. И другие не менее интересные сайты по тематике курса.**

Введение в ПиОА

Включение

поддержки

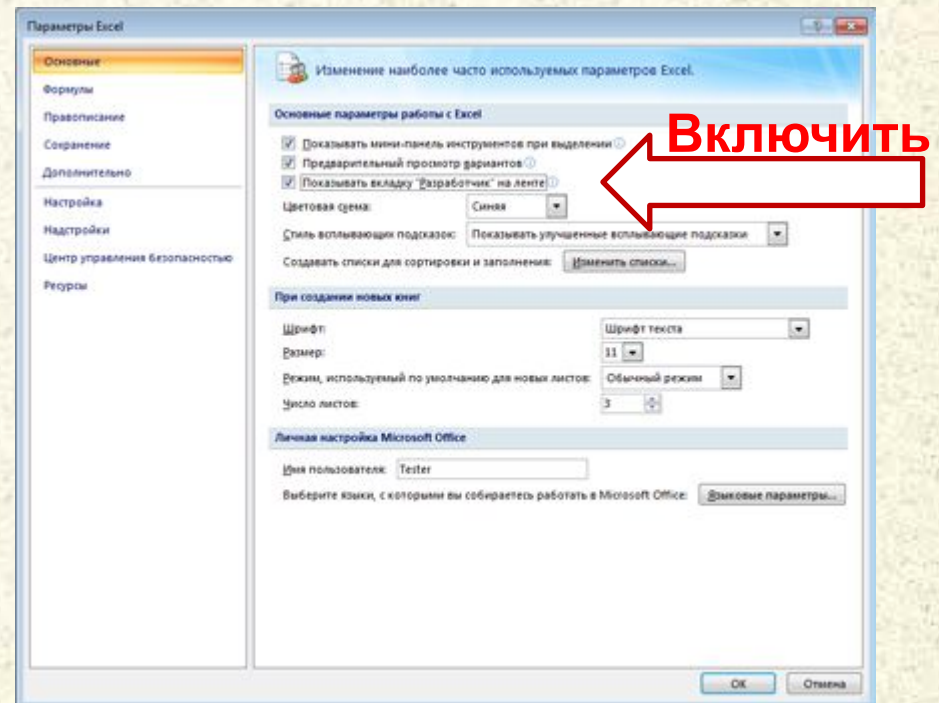
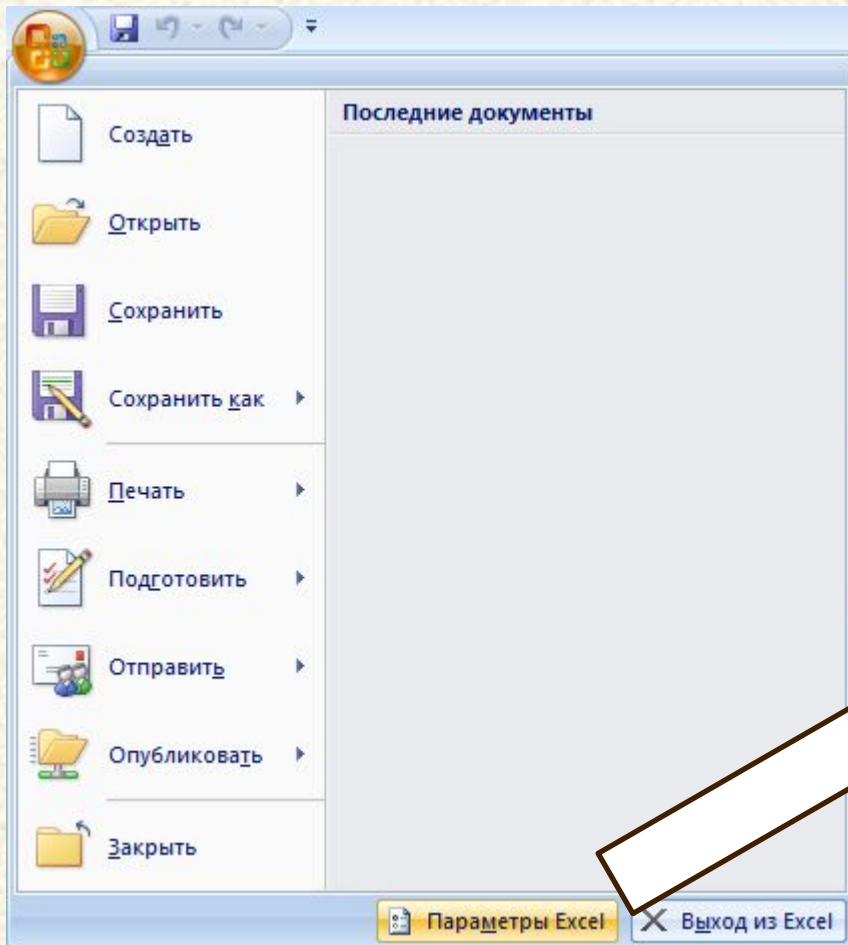
средств

автоматизации

в MsExcel

Включение вкладки «Разработчик»

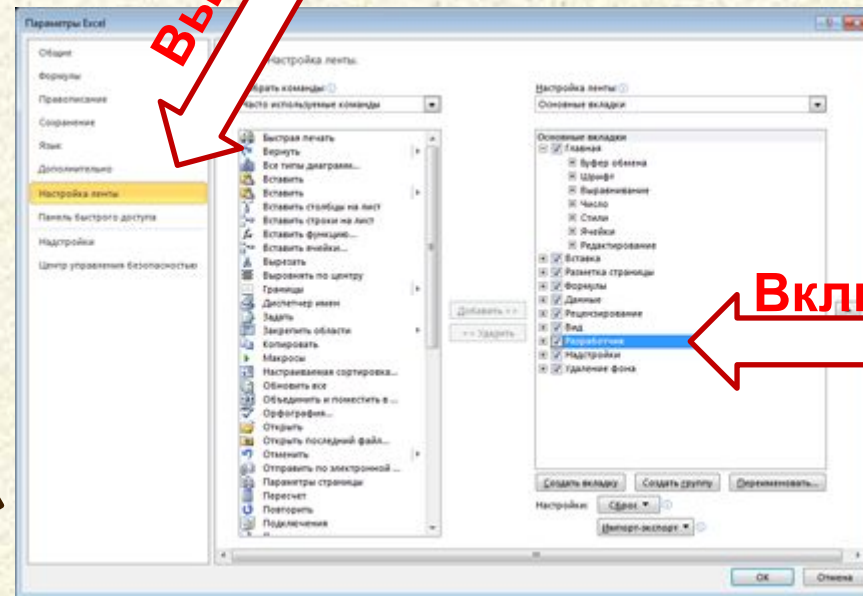
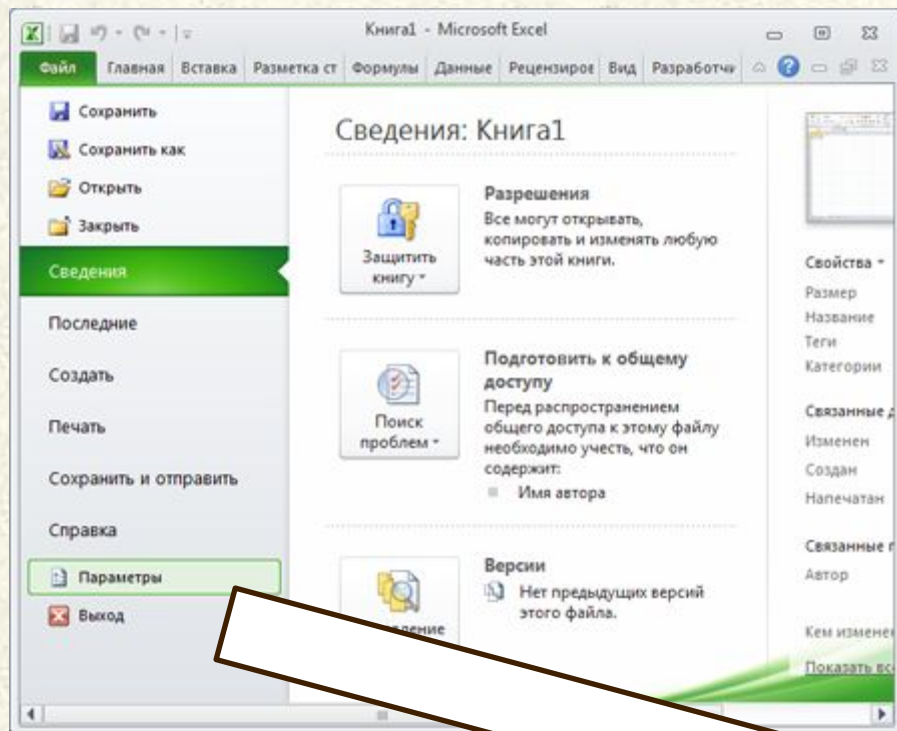
Ms Excel 2007



1. Нажмите кнопку **Microsoft Office** и выберите пункт **Параметры Excel**
2. Откройте вкладку **Основные**, установите флажок **Показывать вкладку "Разработчик" на ленте** и нажмите кнопку **ОК**.

Включение вкладки «Разработчик»

Ms Excel 2010



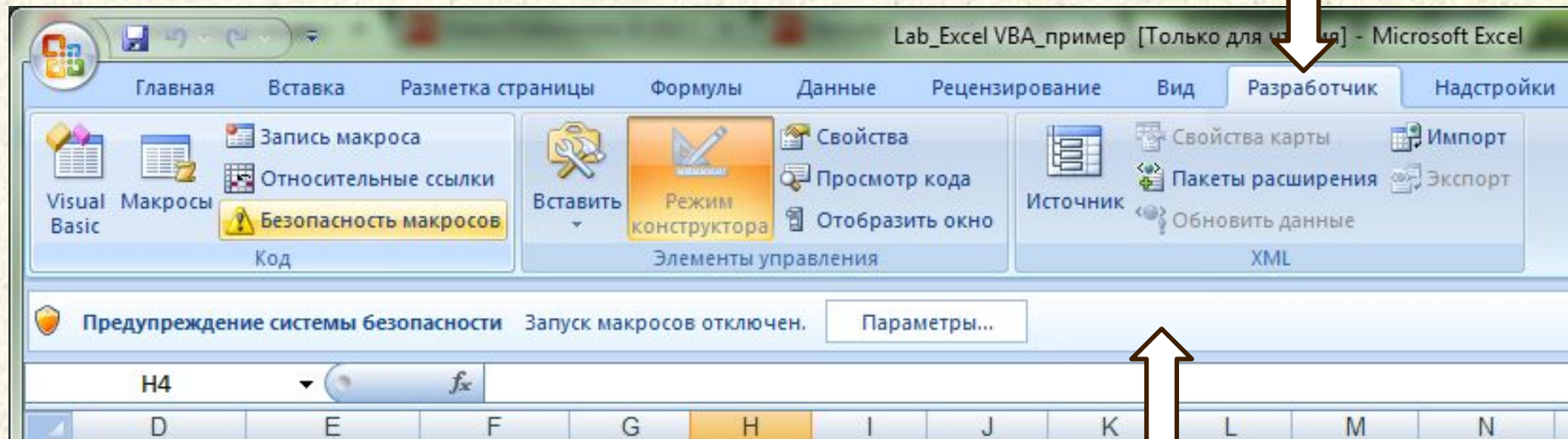
1. На вкладке **Файл** выберите пункт **Параметры**.
2. Выберите **Настройка ленты** на левой панели.
3. Установите флажок **Разработчик** в разделе **Основные вкладки** справа, а затем нажмите кнопку **ОК**.

Панель «Разработчик» для работы макросами

Вкладка «Разработчик» на ленте Ms Excel 2007

Работа с макросами

Создание графического интерфейса



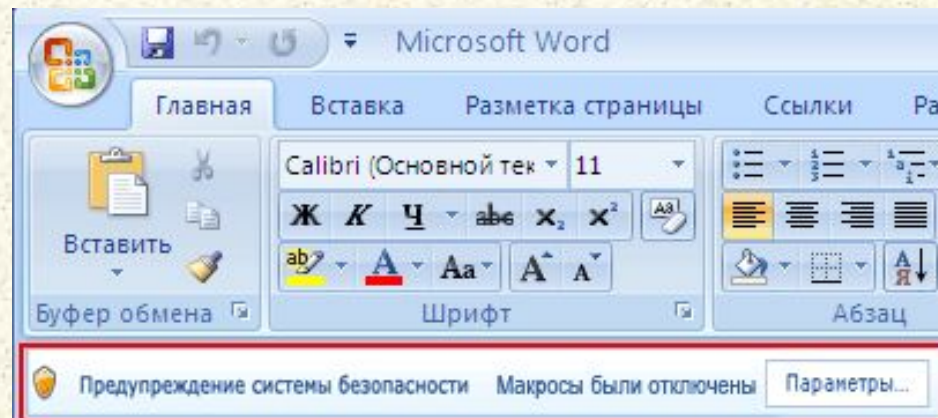
Всплывающая панель системы безопасности

Безопасность при использовании Макросов

Перед включением макроса в документе центр управления безопасностью проверяет следующие сведения:

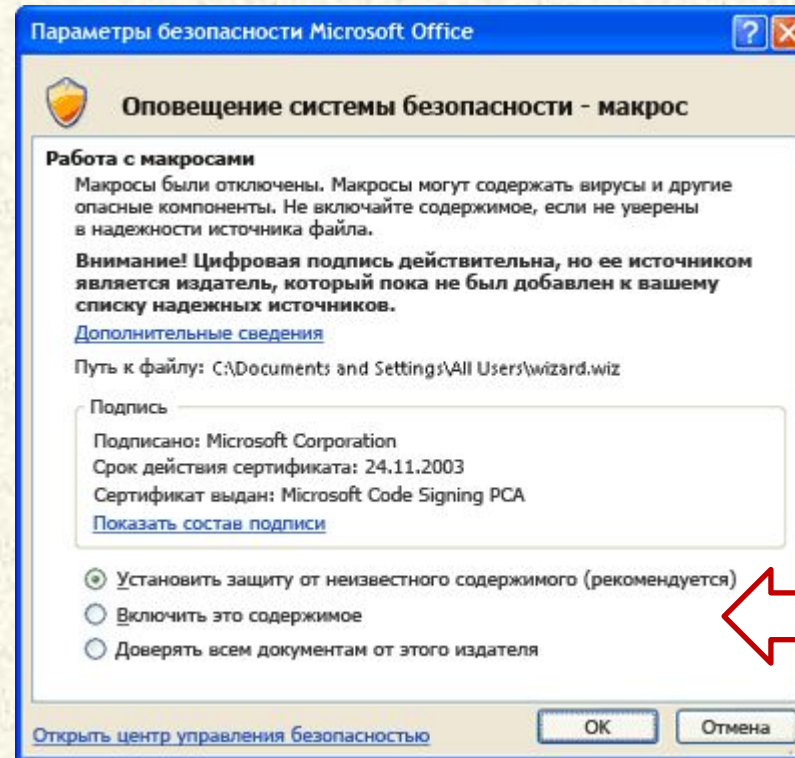
1. макрос должен быть удостоверен цифровой подписью разработчика;
2. цифровая подпись должна быть действительной;
3. цифровая подпись должна быть действующей (не просроченной);
4. сертификат цифровой подписи должен быть выдан общепризнанным центром сертификации;
5. разработчик, удостоверивший макрос, должен быть надежным.

Предупреждение центра безопасности при хотя бы одной проблеме:



Настройка параметров безопасности

Диалоговое окно параметров безопасности позволяет отключить защиту:



1. **Макрос не подписан** – Поскольку макрос не имеет цифровой подписи, разработчик макроса не может быть установлен. Следовательно, невозможно определить, безопасен ли данный макрос.
2. **Цифровая подпись макроса не является надежной** – Макрос потенциально опасен, поскольку хотя и имеет действительную цифровую подпись, но разработчик, подписавший макрос, не является доверенным издателем.
3. **Цифровая подпись макроса недействительна** – Макрос потенциально опасен, поскольку имеет недействительную цифровую подпись.
4. **Цифровая подпись макроса просрочена** – Макрос потенциально опасен, поскольку содержит цифровую подпись, срок действия которой уже истек.

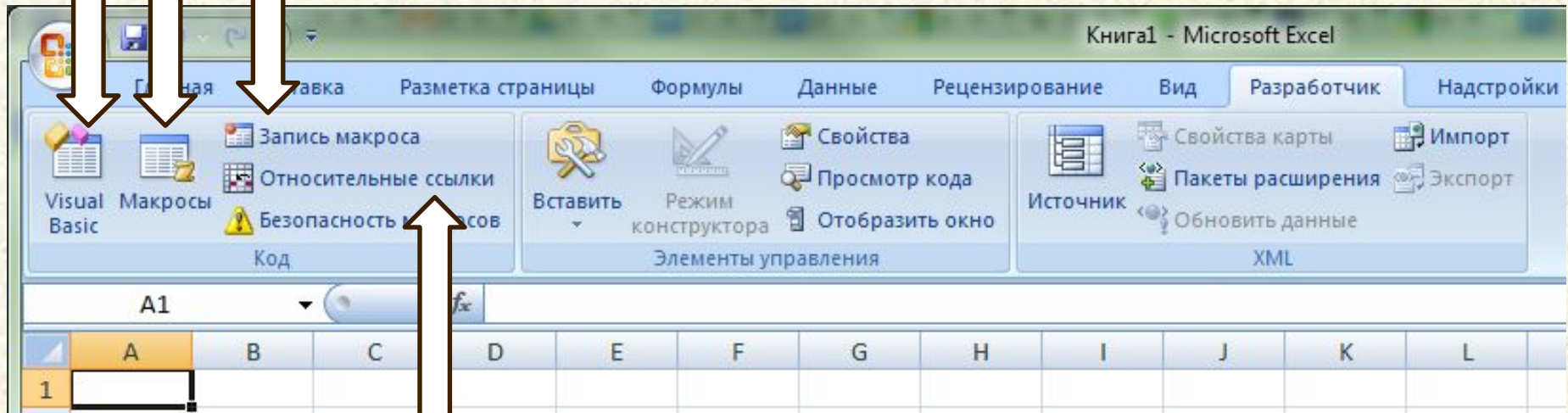
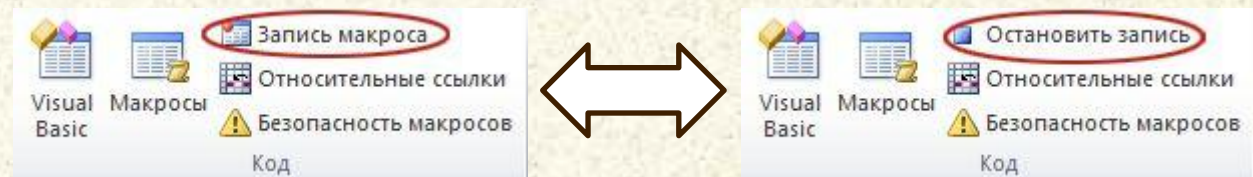
Панель «Разработчик» для работы макросами

Открытие среды разработки VBA

Открытие окна списка макросов

Запись действий пользователя как макрос

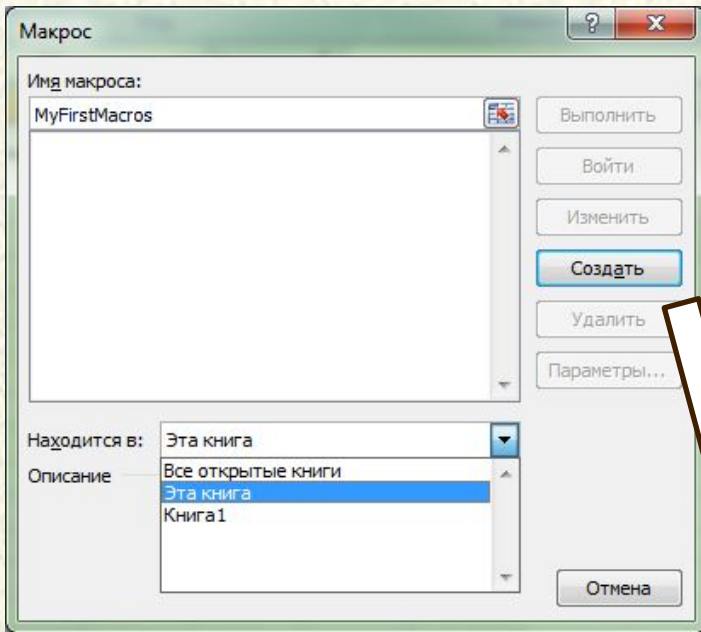
Управление записью макроса



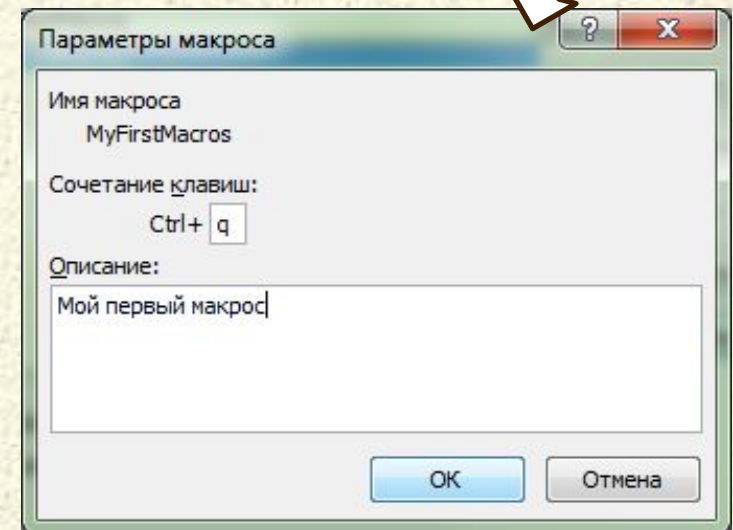
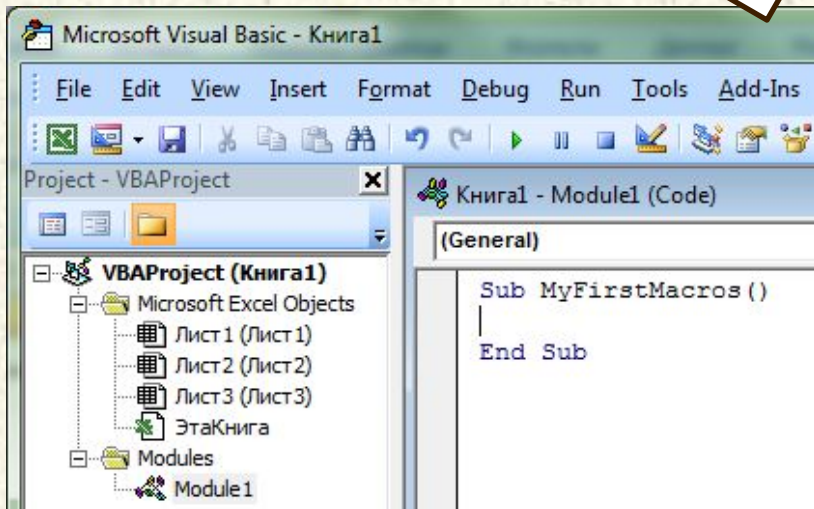
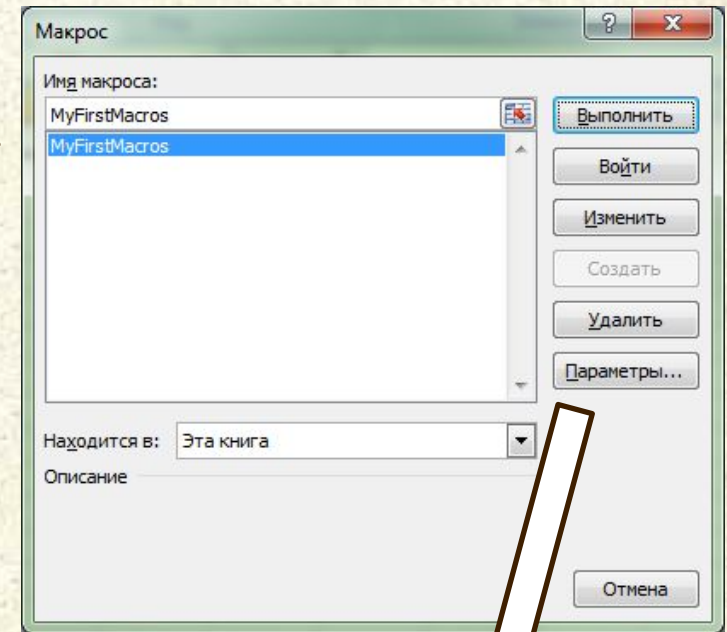
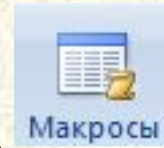
Режим относительных ссылок при переходах

Создание, выполнение и настройка макроса

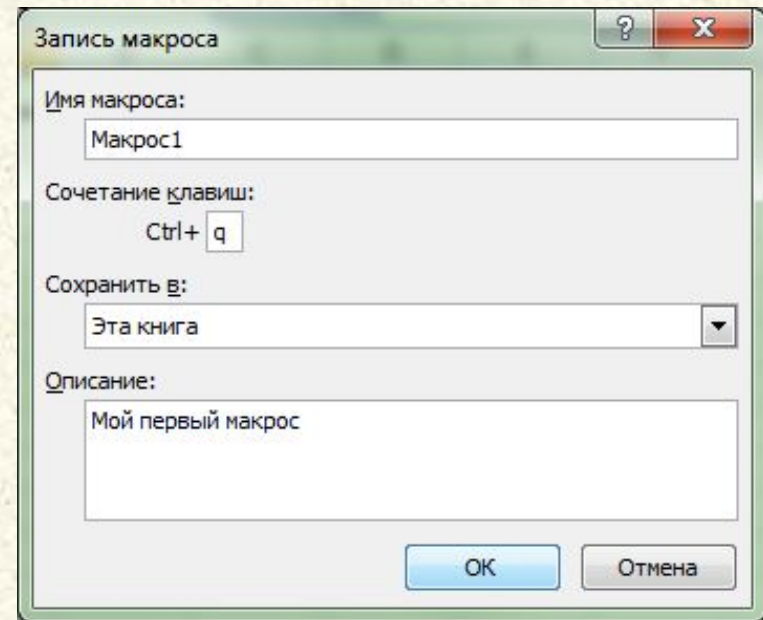
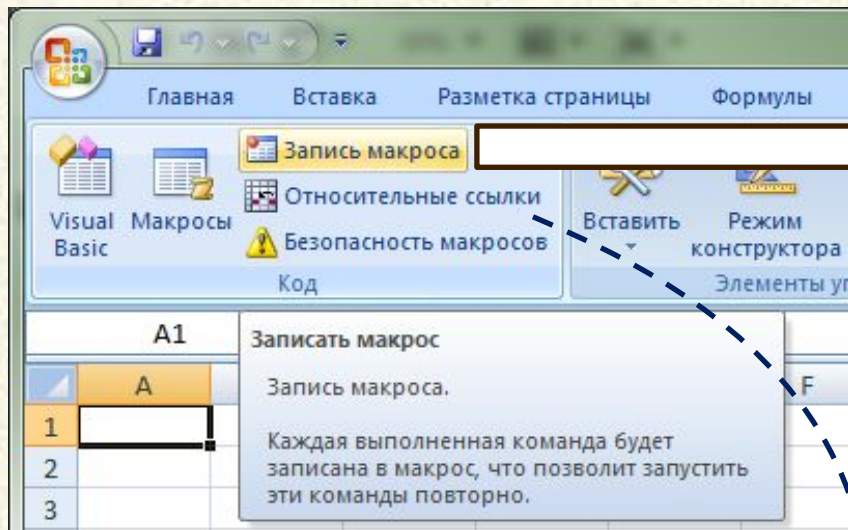
1. Ввод наименования



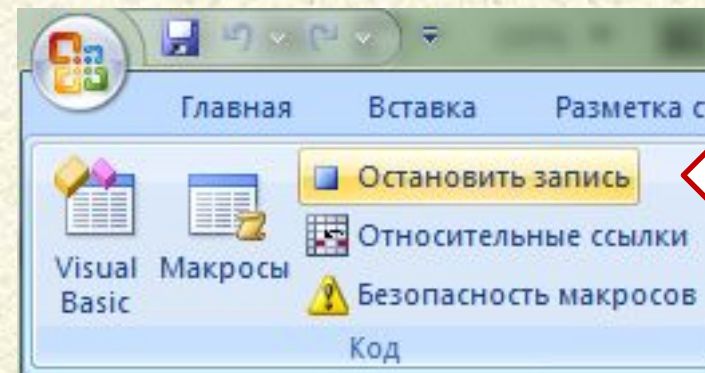
2. Выполнение и настройка



Запись последовательности действий пользователя как макроса



1. Установить курсор в целевую ячейку
2. Выбрать режим ссылок (опция)
3. **Включить запись макроса**
4. Настроить свойства
5. Выполнить запланированную последовательность действий
6. **Завершить запись**



**Нажать
для
окончания
записи**

Пример разработки макроса

Дано: Сводные данные стоимости аренды автомобилей

Цель: Рассчитать стоимость аренды с учётом скидок

Модель	Категория	Валюта	Стоимость автомобиля	Стоимость одного дня проката (\$)	Срок проката	Скидка (\$)	Итого (руб.)
LEXUS GS 300	спорт	usd	73900	100	6		
NISSAN X-Trail	внедорожник	usd	30190	85	4		
HONDA CR-V	спорт	usd	35400	80	5		
FORD Explorer	внедорожник	usd	43600	90	7		
FORD Maverick	внедорожник	usd	33400	90	10		
PEUGEOT 307	город	eur	455500	50	7		
CHRYSLER CROSSFIRE	спорт	eur	57300	90	7		
CITROËN C2	город	usd	15790	40	2		
TOYOTA RAV4	внедорожник	usd	33000	50	3		
TOYOTA COROLLA	город	usd	18100	50	10		
TOYOTA RAV4	внедорожник	usd	33000	60	11		
Alfa Romeo 147 GTA	спорт	eur	47200	100	15		
Volvo XC90	внедорожник	usd	56900	70	4		
NISSAN X-Trail	внедорожник	usd	30190	75	2		
MINI Coopers S	город	eur	29900	55	1		
FORD Focus	город	usd	19560	45	10		
PEUGEOT 407	город	usd	22750	45	6		
PEUGEOT 607	город	usd	32740	70	8		
PEUGEOT 206	город	usd	12500	40	5		
CHRYSLER CROSSFIRE	спорт	eur	57300	80	3		
TOYOTA Land Cruiser Prado	внедорожник	usd	60900	80	3		
PEUGEOT 407	город	usd	2750	40	8		
FORD Focus	город	usd	19560	45	10		
PEUGEOT 307	город	eur	455500	45	3		
CITROËN C2	город	usd	15790	40	9		
TOYOTA Camry	город	usd	37600	60	10		
TOYOTA RAV4-s	внедорожник	usd	35400	50	10		
PORSCHE Cayenne S Tur	внедорожник	eur	165800	120	9		
PORSCHE Carrera4	спорт	eur	177000	120	9		
Итого:							

Текущий курс:		
usd	eur	
26,85	34,41	

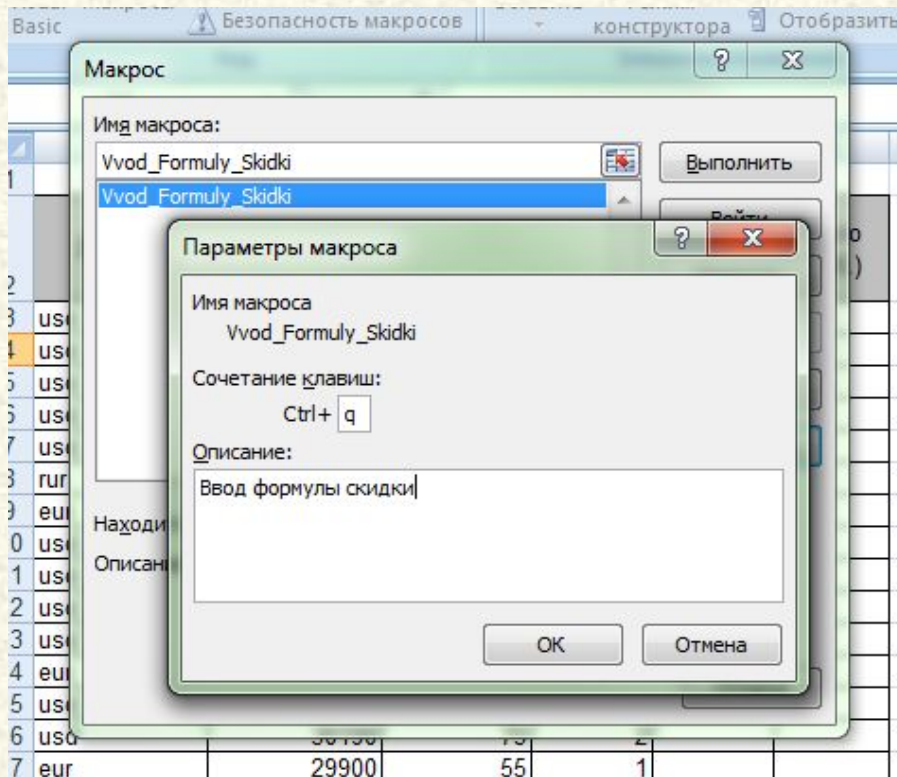
Если количество дней проката больше 7, то предоставляется скидка 10%

Расчет кредита	Тип 1	Тип 2
Начальный взнос	0%	30%
Ставка кредита	11%	6%
Срок погашения (мес)	12	18

Наименование модели			
Стоимость (руб.)			
Ежемесячный платеж (руб.)			
Калькуляция платежей	№ месяца	Остаток 1	Остаток 2

Пример разработки макроса

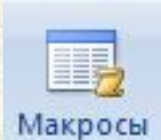
1. Установить курсор в ячейку H3
2. Начать запись макроса с именем Vvod_Formuly_Skidki



=ЕСЛИ(G3>7;G3*F3*0,1;0)				
E	F	G	H	I
Стоимость автомобиля	Стоимость одного дня проката (\$)	Срок проката	Скидка (\$)	Итого (руб.)
73900	100	6	3*0,1;0)	
20100	25	1		

3. Ввести формулу скидки
4. Остановить запись макроса

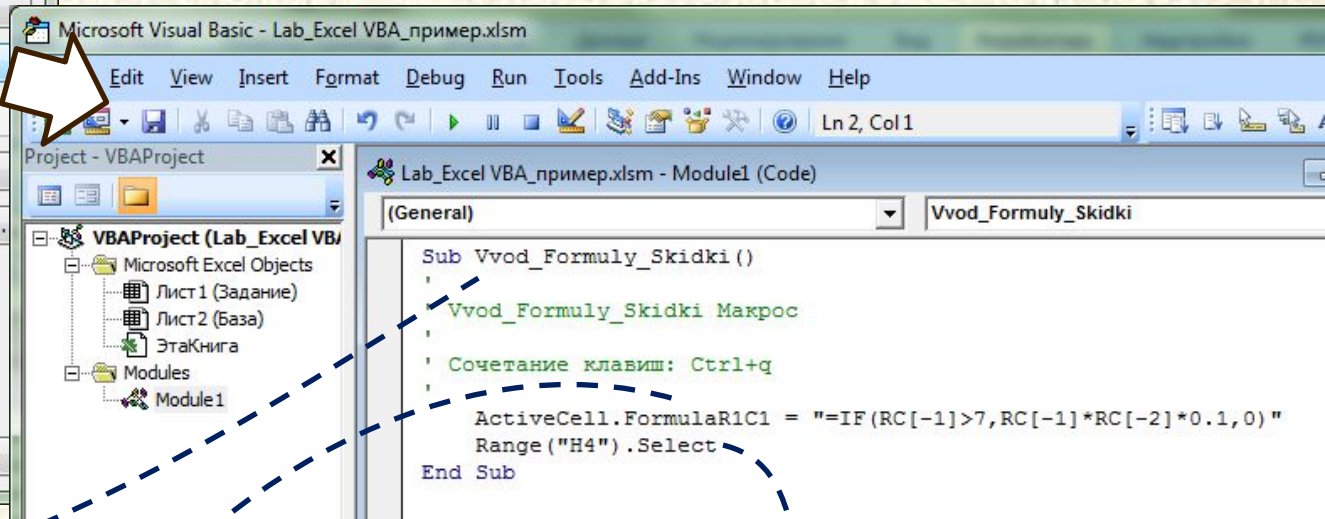
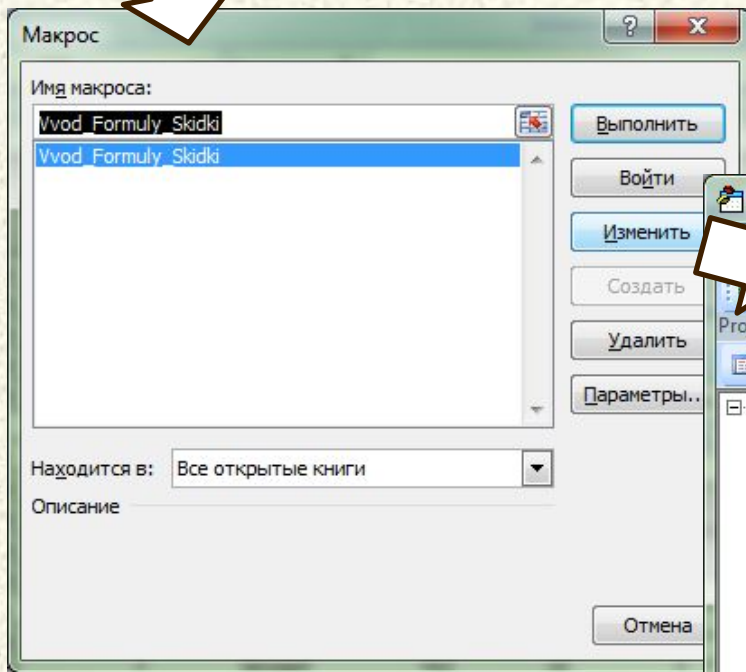
Пример разработки макроса (продолжение)



5. Открыть список макросов

6. Выбрать макрос и нажать кнопку **Изменить**

7. Откроется окно редактора VBA



Название макроса

Формула в стиле «R1C1»

Выделение ячейки «H4»*

*Изменение в режиме «Относительные ссылки»

Сдвиг →

```
Range("H4").Select  
ActiveCell.Offset(1, 0).Range("A1").Select
```


Пример разработки макроса (продолжение)

8. Модифицируем код макроса для заполнения соседней ячейки второй формулой

```
Cells(ActiveCell.Row, ActiveCell.Column).FormulaR1C1 = "=IF(RC[-1]>7,RC[-1]*RC[-2]*0.1,0)"  
Cells(ActiveCell.Row, ActiveCell.Column + 1).FormulaR1C1 = "=(RC[-2]*RC[-3]-RC[-1])*usd"  
Cells(ActiveCell.Row + 1, ActiveCell.Column).Select
```

9. Сохраним изменения и вернемся в Excel

10. Присвоим ячейке содержащей курс доллара (L3) имя «usd»

11. Выполним макрос нажатием **Ctrl + q**

G	H	I	J	K	L	M
					Текущий курс:	
Срок проката	Скидка (\$)	Итого (руб.)			usd	eur
6	0	16110			26,85	34,41
4						
5						

← Нужно следить за включённой раскладкой клавиатуры

12. Повторим нажатие **Ctrl + q** нужное количество раз

... и продолжаем эксперименты самостоятельно !!!

Пример разработки макроса (окончание)

Итого:										Текущий курс:		
№	Модель	Категория	Валюта	Стоимость автомобиля	Стоимость одного дня проката (\$)	Срок проката	Скидка (\$)	Итого (руб.)	Итого	Итого	Итого	Итого
1	LEXUS GS 300	спорт	usd	73900	100	6	0	16110	usd	eur	26,85	34,41
2	NISSAN X-Trail	внедорожник	usd	30190	85	4	0	9129	Если количество дней проката больше 7, то предоставляется скидка 10%			
3	HONDA CR-V	спорт	usd	35400	80	5	0	10740				
4	FORD Explorer	внедорожник	usd	43600	90	7	0	16915,5	Расчет кредита			
5	FORD Maverick	внедорожник	usd	33400	90	10	90	21748,5				
6	PEUGEOT 307	город	eur	455500	50	7	0	9397,5	Тип 1		Тип 2	
7	CHRYSLER CROSSFIRE	спорт	eur	57300	90	7	0	16915,5	Начальный взнос		0% 30%	
8	CITROËN C2	город	usd	15790	40	2	0	2148	Ставка кредита		11% 6%	
9	TOYOTA RAV4	внедорожник	usd	33000	50	3	0	4027,5	Срок погашения (мес)		12 18	
10	TOYOTA COROLLA	город	usd	18100	50	10	50	12082,5	Наименование модели			
11	TOYOTA RAV4	внедорожник	usd	33000	60	11	66	15948,9	Стоимость (руб.)			
12	Alfa Romeo 147 GTA	спорт	eur	47200	100	15	150	36247,5	Ежемесячный платеж (руб.)			
13	Volvo XC90	внедорожник	usd	56900	70	4	0	7518	Калькуляция платежей			
14	NISSAN X-Trail	внедорожник	usd	30190	75	2	0	4027,5	№ месяца		Остаток 1 Остаток 2	
15	MINI Coopers S	город	eur	29900	55	1	0	1476,75				
16	FORD Focus	город	usd	19560	45	10	45	10874,25				
17	PEUGEOT 407	город	usd	22750	45	6	0	7249,5				
18	PEUGEOT 607	город	usd	32740	70	8	56	13532,4				
19	PEUGEOT 206	город	usd	12500	40	5	0	5370				
20	CHRYSLER CROSSFIRE	спорт	eur	57300	80	3	0	6444				
21	TOYOTA Land Cruiser Prado	внедорожник	usd	60900	80	3	0	6444				
22	PEUGEOT 407	город	usd	2750	40	8	32	7732,8				
23	FORD Focus	город	usd	19560	45	10	45	10874,25				
24	PEUGEOT 307	город	eur	455500	45	3	0	3624,75				
25	CITROËN C2	город	usd	15790	40	9	36	8699,4				
26	TOYOTA Camry	город	usd	37600	60	10	60	14499				
27	TOYOTA RAV4-s	внедорожник	usd	35400	50	10	50	12082,5				
28	PORSCHE Cayenne S Tur	внедорожник	eur	165800	120	9	108	26098,2				
29	PORSCHE Carrera4	спорт	eur	177000	120	9	108	26098,2				
30									Итого:			

... и продолжаем эксперименты самостоятельно !!!

Ключевые моменты кода макроса

Начало и конец кода макроса, **sub** от subprogram (подпрограмма)

Наименование макроса (может быть записано кириллицей)

Sub Vvod_Formuly_Skidki()

Комментарий к тексту программы

' Макрос записан 1.12.2012

' Быстрый вызов Ctrl + Q

Объект – активная ячейка (выделенная)

Cells(ActiveCell.Row, ActiveCell.Column).FormulaR1C1 = "=IF(RC[-1]>7,RC[-1]*RC[-2]*0.1,0)"

Cells(ActiveCell.Row, ActiveCell.Column + 1).FormulaR1C1 = "=(RC[-2]*RC[-3]-RC[-1])*usd"

Cells(ActiveCell.Row + 1, ActiveCell.Column).Select

End Sub

Массив (коллекция) ячеек рабочего листа
(сокращенная запись)

Полная форма записи:

Application.Workbooks(1).Worksheets(1).Cells(Row, Column)

ActiveWorkBook – активная рабочая книга

ActiveSheet – активный лист
текущей рабочей книги

Структурные диаграммы

**Элементы и
правила
построения
блок-схем**

Определение

Блок-схема является формой представления алгоритма с помощью графических символов. Графические символы, их размеры, а также правила построения блок-схем определены государственными стандартами.

ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

ГОСТ 19.002-80. Схемы алгоритмов и программ. Правила выполнения.

ГОСТ 19.003-80. Схемы алгоритмов и программ. Обозначения условные графические

Элементы блок-схем (международная традиция)

Процесс. Выполнение операции или группы операций, в результате чего изменяется значение, форма представления или расположения данных. Внутри символа или же в виде комментария на естественном языке или в виде формулы записываются действия, которые производятся при выполнении операции или группы операций.



Предопределенный процесс. Использование ранее созданных и отдельно описанных алгоритмов или программ (процедур, функций, программных модулей). Символ служит для указания обращения к процедурам, функциям, программным модулям.



Элементы блок-схем (продолжение)

Решение. Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий.

Символ используется для изображения унифицированных структур:

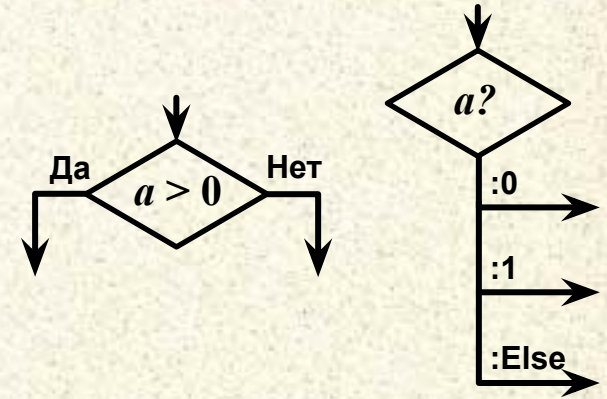
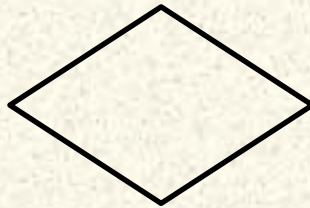
РАЗВИЛКА ПОЛНАЯ

РАЗВИЛКА НЕПОЛНАЯ

ВЫБОР

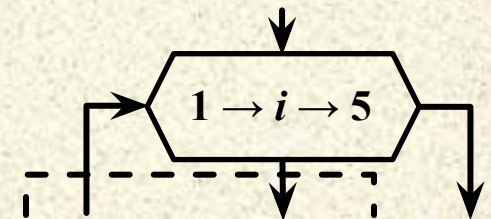
ЦИКЛ-ДО

ЦИКЛ-ПОКА



Модификация. Выполнение операций, меняющих команды или группу команд, изменяющих программу.

Символ используется для изображения унифицированной структуры ЦИКЛ С ПАРАМЕТРОМ. Внутри символа записывается параметр цикла с указанием начального и конечного значений, а также шаг изменения цикла, если он не равен единице.



Элементы блок-схем (продолжение)

Ввод-вывод . Операция обмена данными с внешним устройством хранения, ввода-вывода, базой данных, элементами управления или временным хранилищем в оперативной памяти.



Ручной ввод. Ввод данных оператором в процесс обработки при помощи устройства, непосредственно сопряженного с компьютером (например, клавиатура).



Дисплей. Ввод-вывод данных в случае, если непосредственно подключенное к процессору устройство воспроизводит данные и позволяет оператору вносить изменения в процессе их обработки.

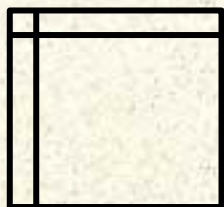


Элементы блок-схем (продолжение)

Карта. (*устар.*) Ввод-вывод данных с использованием в качестве носителя малой ёмкости, стандартизированного дизайна и, как правило, однократной записи – перфоркарты, RFID-метик или штрих-кода.



Память. Ввод - вывод данных в хранилище, организованное в оперативной памяти или на ином устройстве, допускающим произвольный доступ



Документ. Ввод - вывод данных, носителем которых служит бумага.



Лента. (*устар.*) Ввод-вывод данных в «человеко-читаемом» виде на носитель или устройство, имитирующее бумажную ленту (перфоленту, кассовую ленту, стример, «бегущую строку», последовательный канал передачи данных).



Элементы блок-схем (продолжение)

Линия потока. Указание последовательности связей между символами.



Правила изображения линий потока:

- 1) линии потока должны быть параллельны линиям внешней рамки блок-схемы (границам листа, на котором изображена блок-схема);
- 2) направление линии потока сверху вниз и слева направо принимается за основное и стрелками не обозначается, в остальных случаях направление линии потока обозначается стрелками;
- 3) изменение направления линии потока производится под углом 90 градусов;
- 4) слияние (объединение) линий (кроме тривиальных случаев) должно обозначаться **узлом** или **концевыми стрелками** на линиях потоков.

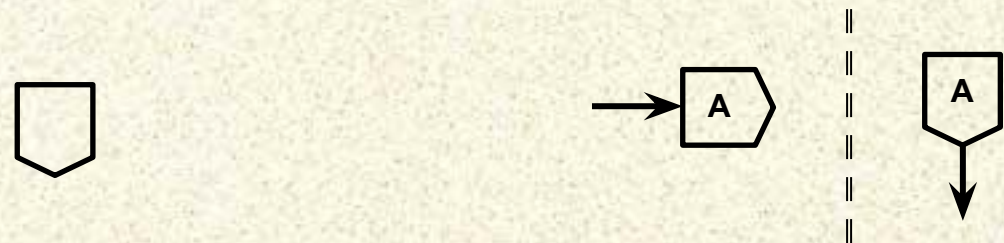


Элементы блок-схем (продолжение)

Соединитель. Указание связи между прерванными линиями потока, связывающими символы. Если блок-схема состоит из нескольких частей, расположенных на одной странице, то линия потока одной части заканчивается символом СОЕДИНИТЕЛЬ, а линия потока на продолжении блок-схемы начинается с этого же символа. Внутри символов СОЕДИНИТЕЛЬ ставятся одинаковые порядковые номера, соответствующие разорванной линии потока

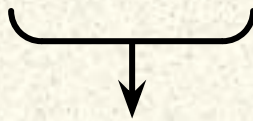


Межстраничный соединитель. Указание связи между разъединенными частями схем алгоритмов и программ, расположенных на разных листах. Данный символ служит для тех же целей, что и соединитель, но при расположении частей блок-схемы на разных страницах.



Элементы блок-схем (окончание)

Пуск - останов. Начало, конец, прерывание процесса обработки данных или выполнения программы.



Комментарий. Связь между элементами схемы и пояснениями. Позволяет включать в блок-схему пояснения, формулы и другую информацию.



Элементы языка VBA

**Особенности
организации
кода**

Общие принципы

Visual Basic for Application – императивный язык программирования высокого уровня.

- Линейное:** Инструкции разделены и следуют последовательно.
- Структурное:** Блоки инструкций не содержат пересечений линий исполнения и собираются по принципу вложенности.
- Процедурное:** Часто повторяющиеся блоки инструкций могут быть выделены в подпрограммы (процедуры и функции) с одним входом и одним выходом.
- Модульное:** Часть кода приложения может быть вынесена в модули (библиотеки), подключаемые при необходимости использования.

Visual Basic for Application – язык с поддержкой объектно-ориентированного и событийного программирования

- Объект:** Совокупность **данных**, характеризующих его состояние, и **функций** их обработки, моделирующих поведение объекта.
- Класс:** Программный шаблон, на основе которого создается объект (реализация).
- Метод:** Функция или процедура, являющаяся частью описания объекта, предназначенная для выполнения каких-либо действий над объектом (данными).
- Событие:** Информационный эквивалент реакции системы на полученное сообщение.
- Обработчик:** Метод объекта, предназначенный для обработки специфического события.

Основной принцип организации кода в **Visual Basic for Application** – модульно-процедурный

Размещение кода: Модули в файлах .bas или в составе документа.

Организация кода: Исполняемые инструкции (команды) размещаются в теле процедур (Sub ... End Sub) или функций (Function ... End Function)

Объявление: В области General и внутри процедур и функций

Видимость: Доступность объекта или процедуры (функции) определяется размещением (например, в том же модуле, в той же процедуре и т.п.)

Модуль ≈ класс объекта (ООП)

Private, public управляют доступом к свойствам (глобальным переменным модуля) и методам (процедурам и функциям)

Элементы языка VBA

Операции с данными Хранение и преобразование

Представление данных в коде

Литералы

Литеральная константа – это данные (число, строковое выражение, дата и т.п.), размещенные непосредственно в коде программы

Константы

Схемы алгоритмов и программ. Правила
Именованная константа – это какие-либо данные, которые не изменяются при выполнении программы, и для обращения к ним используется специальное символьное имя определенное в коде

Допускается использование **типизированных констант** при объявлении которых явно задаётся тип данных

Переменные

Именованные объекты, предназначенные для временного хранения изменяемых данных

Примеры представления данных в коде

Литералы

Литеральная константа – это данные (число, строковое выражение, дата и т.п.), размещенные непосредственно в коде программы

Константы

Схемы алгоритмов и программ. Правила
Именованная константа – это какие-либо данные, которые не изменяются при выполнении программы, и для обращения к ним используется специальное символьное имя определенное в коде

Допускается использование **типизированных констант** при объявлении которых явно задается тип данных

Переменные

Именованные объекты, предназначенные для временного хранения изменяемых данных

Специальные константы

Внутренняя константа – это именованная константа, которая была определена разработчиками VBA.

Внутренние константы для работы с host-приложениями.
Excel содержит внутренние константы для использования с рабочими книгами электронных таблиц.

Word содержит внутренние константы для работы с документами и шаблонами текстового редактора

Access – константы для операций с базами данных.

Внутренние константы, определяемые VBA, начинаются с букв **vb**.

Внутренние константы Excel - **xl**; Word - **wd**.

Полный список имеющихся в наличии внутренних констант доступен через **Object Browser**.

Его можно вызвать клавишей F2 в окне редактора VBA.

Использование литералов и констант

Начало и конец кода макроса

Наименование макроса (может быть записано кириллицей)

Комментарий к тексту программы

```
Sub Vvod_Formuly_Skidki()
```

```
' Макрос записан 1.12.2012
```

```
' Быстрый вызов Ctrl + Q
```

Объявление типизированной константы

```
Const A As Integer = 1
```

Строковый литерал – последовательность символов в коде программы, интерпретируемых как данные

```
ActiveCell.Value = "Hello world"
```

Числовой литерал

```
Cells(ActiveCell.Row, ActiveCell.Column + 1).FormulaR1C1 = "(RC[-2]*RC[-3]-RC[-1])*usd"
```

```
Cells(ActiveCell.Row + A, ActiveCell.Column).Select
```

```
End Sub
```

Использование константы

Метод

Объект

Свойство

Объявление переменных

Оператор **Dim** (от **dimention**) – Объявляет и размещает в памяти одну или несколько переменных.

Область
General
модуля

Dim i

Объявление нетипизированной
глобальной переменной

Dim j As Integer

Объявление типизированной
глобальной переменной

Sub MySub()

' Макрос записан 1.12.2012

' Быстрый вызов Ctrl + Q

Объявление типизированной
переменной и инициализация
значением

Dim X As Double = 0.5

Неявное объявление через
установку (присвоение)
значения

Y = 1 + X

Использование переменной

ActiveCell.Value = "1" + Y

Неявное преобразование типов

End Sub

Оператор присваивания

Типы данных

Тип данных	Резервируется байт	Наименьшее значение	Наибольшее значение
Byte	1	0	255
Boolean	2	False (Ложь)	True (Истина)
Integer	2	-32768	32767
Long	4	-2147483648	2147483647
Single	4	-3.402823 E38	-1.401298 E-45
		1.401298E-45	3.402823E38
Double	8	-1,79769313486232E308	-4,94065645341247E-324
		4,94065645841247E-324	1,79769313486232E308
Currency	8	-922337203685477,5808	922337203685477,5807
Decimal	14	+/-79228162514264337593543950335 без десятичных знаков	+/-7,9228162514264337593543950335 с 28-ью знаками после запятой
Date	8	1 января 100 года	31 декабря 9999 года
Object	4	Любая ссылка на объект	
String (переменной длины)	10 байт + длина строки	0	приблизительно 2 млрд
String (фиксированной длины)	Длина строки	1	65400
Variant (числа)	16	Любое числовое значение в рамках диапазона типа данных Double	
Variant (символы)	22 байта + длина строки	0	приблизительно 2 млрд
Пользовательский	Зависит от типа	Зависит от элемента	

«Наследие Basic» - явное указание типа

Специальный символ (из не разрешенных к использованию в именах переменных) выступает **указателем типа** переменной при объявлении и использовании.

Dim X# ← Использование указателя типа переменной

Dim Y As Double ← Объявление типизированной переменной

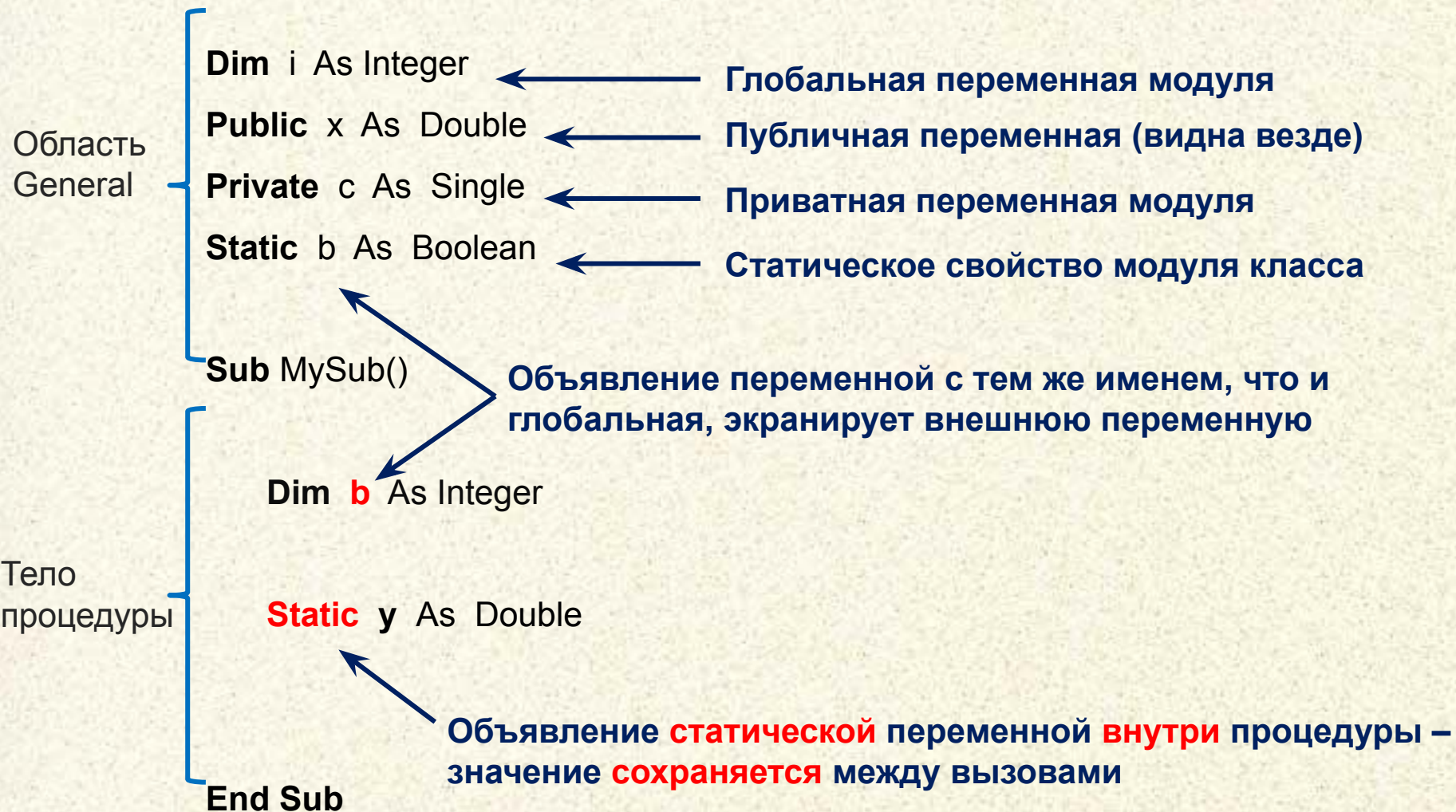
Символы объявления типов:

Тип данных	Символ объявления типа
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	S

Операции с переменными

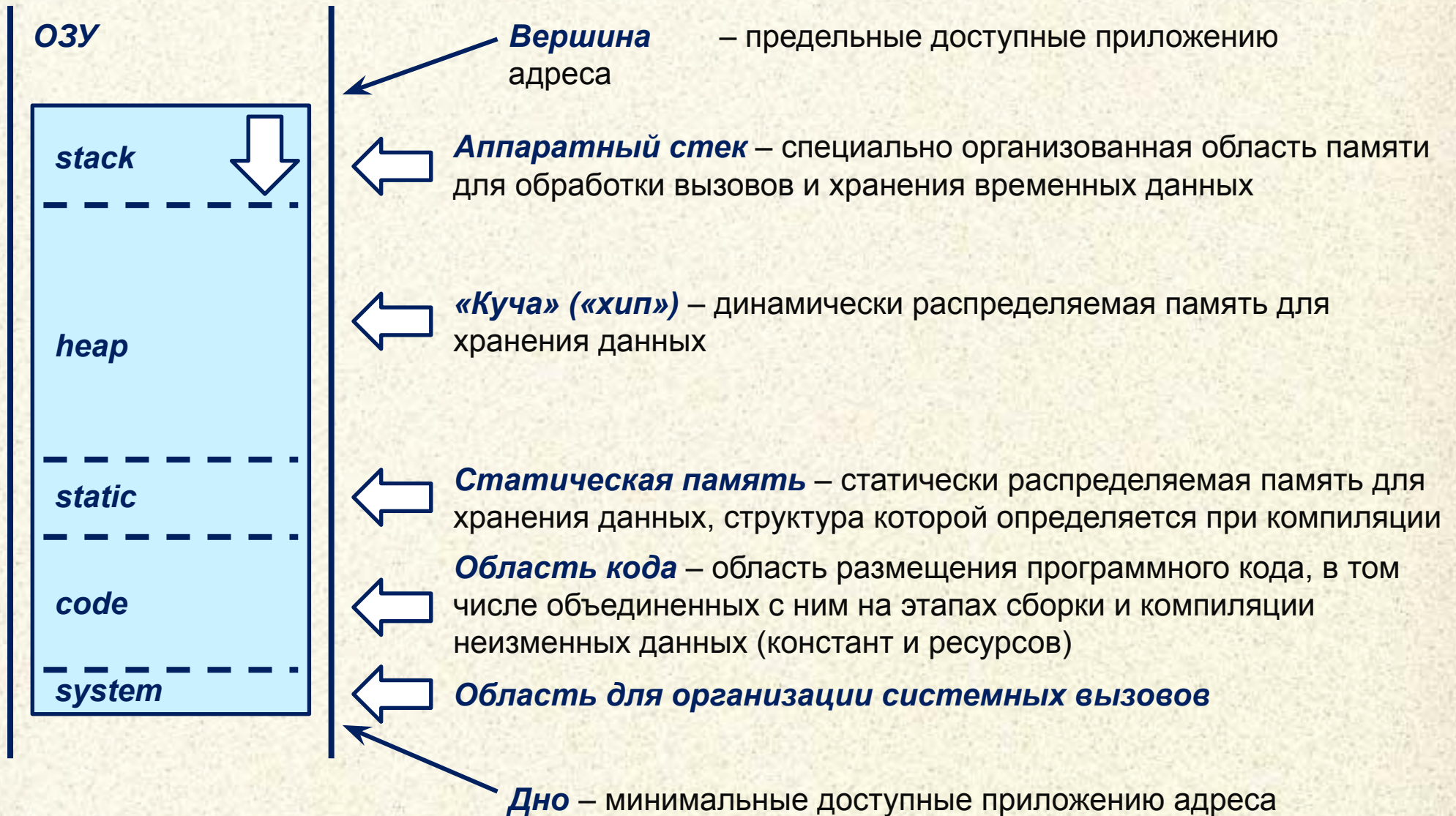
Применение модификаторов области видимости и способа размещения в памяти

- public** – переменная доступна во всех модулях приложения
- private** – переменная доступна только в данном модуле (аналогично **Dim** в **General**)
- static** – переменная размещается в статической памяти



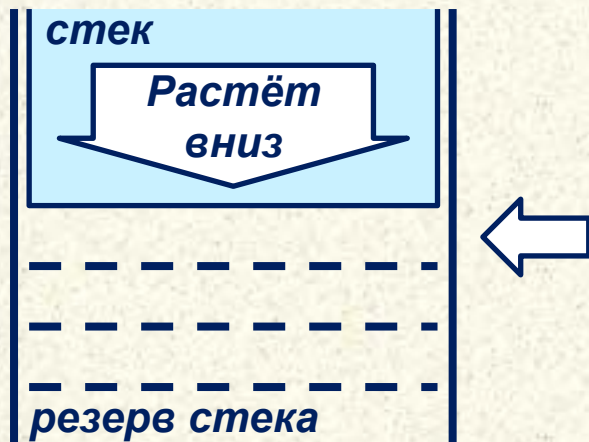
Структура памяти приложения

Размещение в памяти зависит от способа объявления переменной и типа данных



Использование стека при вычислениях

При использовании стека **необходимо следить** за используемыми размерами типов данных



В стековых языках программирования стек используется для размещения данных, а адреса ячеек рассчитываются относительно вершины стека

Вершина стека – указатель на первую свободную ячейку

Адрес вершины стека хранится в специальных регистрах процессора **SS** (селектор регистра стека) и **ESP** (указатель стека)

Инструкция **Push** (Втолкнуть) используется для внесения в стек промежуточных данных и ссылок, автоматически уменьшает **ESP**

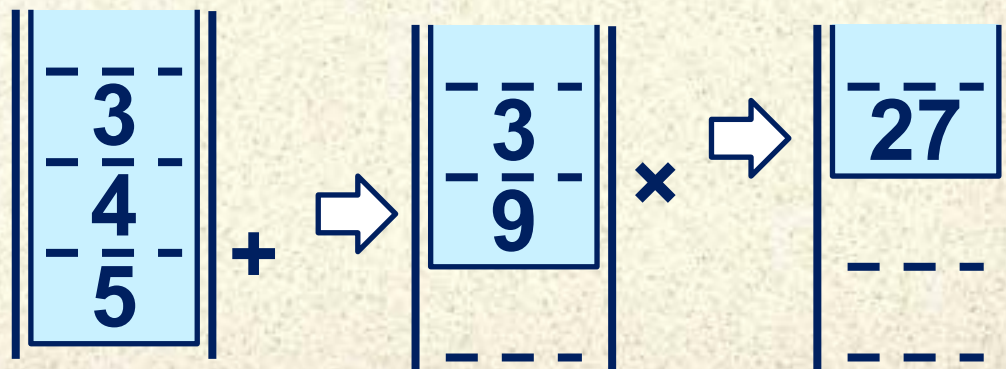
Инструкция **Pop** (Извлечь) используются для получения из стека промежуточных данных и ссылок, автоматически увеличивает **ESP**

Обратная польская нотация позволяет записать математическое выражение в виде последовательности данных (операндов) отдельных бинарных и унарных операций

$$3 \times (4 + 5)$$



$$3 \ 4 \ 5 \ + \ \times$$



Поддержка стековых операций процессором

Базовое использование стека – сохранение состояния процессора (всех регистров) при **вызове подпрограммы** или **обработке прерывания**



Инструкция **Call** (Вызвать) выполняет сохранение **всех регистров** стек, затем в стек помещается **адрес выборки команды**, а потом выполняется **переход по адресу**, указанному в инструкции

Interrupt (Прерывание) – встроенная последовательность операций процессора, выполняемая в ответ на заданное событие. При обработке прерывания в стек помещается текущий **адрес выборки команды**, затем по номеру прерывания вычисляется ячейка в **таблице прерываний**, содержащая **адрес процедуры обработчика** и выполняется переход по этому адресу

В стековых языках программирования стек используется для размещения данных, а адреса ячеек рассчитываются относительно вершины стека

Стековое размещение переменных

При использовании стека **необходимо следить** за используемыми размерами типов данных



В стековых языках программирования стек используется для размещения данных, а адреса ячеек рассчитываются относительно вершины стека

Вершина стека – указатель на первую свободную ячейку

Адрес вершины стека хранится в специальных регистрах процессора **SS** (селектор регистра стека) и **ESP** (указатель стека)

Инструкция **Push** (Втолкнуть) используется для внесения в стек промежуточных данных и ссылок, автоматически уменьшает **ESP**

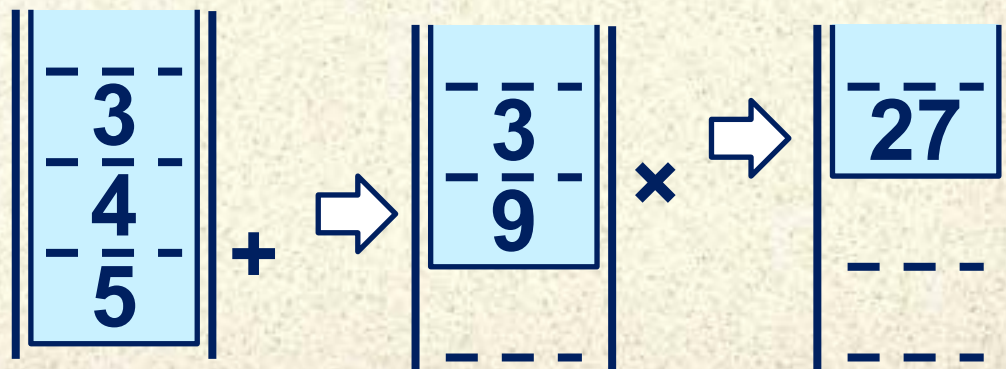
Инструкция **Pop** (Извлечь) используются для получения из стека промежуточных данных и ссылок, автоматически увеличивает **ESP**

Обратная польская нотация позволяет записать математическое выражение в виде последовательности данных (операндов) отдельных бинарных и унарных операций

$$3 \times (4 + 5)$$



$$3 \ 4 \ 5 \ + \ \times$$



Элементы языка VBA

**Основные
алгоритмические
конструкции**

МГМУ «МАМИ»

Кафедра «Автоматика и процессы управления»

Блок дисциплин

Информатика и информационные технологии

Спасибо за внимание !!!

Контакты:

mami.testolog.ru

timid@mami.ru

inform437@gmail.com