

# Лекция №13

- *Языки программирования, их классификация.*
- *Макропрограммирование. Способы создания макропрограмм.*
- *Знакомство со средой программирования VBA. Программирование в среде VBA.*
- *Понятие модуля. Процедуры и способы их создания в VBA.*
- *Примеры создание процедур – подпрограмм и процедур – функций с помощью VBA.*

# ***Языки программирования, их классификация.***

***Язык программирования*** – совокупность средств и правил для представления алгоритма решения задачи в виде, понятном для компьютера.

Существует разделение языков на две группы:

- **Языки высокого уровня;**
- **Языки низкого уровня.**

Языком самого низкого уровня является язык машинных кодов. К этой группе языков относятся **Ассемблер.**

**Достоинства:** они почти не требуют перевода для компьютера, и поэтому быстро выполняются.

**Недостаток:** писать программы на Assembler могут только специалисты.

Языки высокого уровня, такие как **Фортран, Pascal, Си, Си ++, Ада, Basic** – языки программирования для решения конкретных задач.

Программа, написанная на этих языках, требует перевода на язык машинных кодов.

В последние годы широкое распространение получили объектно-ориентированные языки программирования: с приставкой **Visual**

В Excel интегрирована неполная версия языка программирования **Visual Basic** под названием **Visual Basic for Applications (VBA)**.

Отличие VB от VBA в следующем:

- **VB** создаёт законченные самостоятельные приложения, **VBA** – автоматизация существующих приложений.
- **VB** имеет собственную среду разработки, а **VBA** использует среду Word, Excel, Access и т.д. .
- Выполнение – **VBA**-приложений требует Word, Excel, Access и т.д., **VB** – нет.

# Создание макросов

**Макрос** – записанная последовательность команд и действий пользователя, сохранённая под уникальным именем, которую может выполнить Excel.

Макрос можно создавать 2-мя способами:

- в режиме «ЭХО»
- с использованием редактора **VBA**

Макрос используется для автоматизации выполнения часто повторяющихся операций.

# Создание макроса в режиме «ЭХО»

- **Сервис, Макрос, Начать Запись;**
- В появившемся окне необходимо указать:
  - ✓ **имя макроса,**
  - ✓ **назначить, при необходимости, сочетание клавиш для запуска макроса,**
  - ✓ **указать место хранения макросов и щелкнуть Ок.**
- Далее необходимо выполнить те действия, которые мы хотим записать в макрос.

Во время записи макроса на экране присутствует панель инструментов управления записью макроса с одной кнопкой – остановка.

- Для окончания записи макроса щелкните мышью по этой кнопке или введите команду **Сервис, Макрос, Остановить запись.**

При использовании макросов в Excel необходимо сразу позаботиться о безопасности от заражения вирусами, распространяющихся через макросы программ Word, Excel и другие приложения Windows.

В Excel имеется три степени защиты от вирусов: **низкая, средняя и высокая.**

Для работы в сети рекомендуется устанавливать **среднюю** степень защиты. Для этого необходимо выполнить команду

**Сервис, Макрос, Безопасность** и установить переключатель для **средней** степени защиты.

## Выполнение макроса.

**Сервис→Макрос→Макросы→Имя макроса →  
Выполнить**

Можно присвоить макросу:

- *Пункт меню*
- *Кнопку на панели инструментов или командную кнопку*
- *Сочетание клавиш*

Проще всего сочетание клавиш:

***Сервис →Макрос→Начать запись→ctrl+?***

**ИЛИ**

***Сервис→макрос→макросы→имя  
макроса→параметры.***

# Назначение макросов командным кнопкам.

- *Вид – Панели инструментов – Формы*
- Выберите курсором на панели инструментов *Формы* элемент управления *Кнопка*
- Нарисуйте кнопку левой клавишей мыши. Появляется окно *«Назначить макрос объекту»*.
- Выберите нужный макрос и нажмите ОК.
- Измените название *Кнопка1* на подходящее для макроса название.

# Назначение макросов кнопкам панелей инструментов.

- **Сервис** → **Настройка** → **Команды** →  
→ **Категории** → **Макросы**
- В списке **Команды** выберите –  
**Настраиваемая кнопка.**
- Перетащить кнопку на панель инструментов
- Правая клавиша мыши на кнопке и выбрать значок для кнопки
- Правая клавиша и выбрать **Назначить макрос.**

# Программирование на VBA.

Коды в VBA хранятся в модулях. Различают 2 основных типа модулей:

1) *модули классов.*

2) *стандартные модули.*

**Определение модуля:** Модуль определяется как совокупность объявлений (описательная часть) и процедур, хранящихся как единое целое.

**Модули классов** содержат описание собственных объектов, которые можно создавать с помощью VBA.

**Стандартные модули** (или просто модули) содержат либо процедуры-подпрограммы, либо процедуры-функции.

**Процедура** – любая совокупность кода VBA, рассматриваемая как единое целое.

Как правило, процедура состоит из операторов, выполняющих какую-либо задачу или вычисляющих значение.

Каждая процедура идентифицируется своим уникальным именем.

Существуют 2 типа процедур:

- **процедуры-подпрограммы,**
- **процедуры-функции.**

Подпрограмма выполняет один или несколько операторов и **не возвращает значения**.

Примером подпрограммы является код макроса.

# Создание процедуры.

Создание одной процедуры требует выполнения двух последовательных шагов:

- Вставить модуль в рабочую книгу
- В созданном модуле написать процедуру.

## **Последовательность шагов.**

- Создать новую рабочую книгу.
- Сервис — Макрос — Редактор VBA.
- В левой части окна редактора VB расположено окно проектов **Project Explorer** (если его нет, то нужно CTRL+R). В этом окне откроете проект VBA PROJECT (Книга 1) и щёлкните правой клавишей мыши на элементе **Эта книга**.
- В контекстном меню выберите команду **Insert – Module**. Будет добавлен модуль в наше приложение.
- Выполните команду **Insert – Procedure**. Откроется окно Add Procedure.(См. пример!)

# Существуют следующие типы области видимости:

- 1) По умолчанию **процедуры VBA** (за исключением процедур обработки событий) рассматриваются как открытые (**Public**). Это означает, что вы можете вызвать их из любой части программы.
- 2) Локальная (**Private**) процедура будет видима, если вы пожелаете, только внутри содержащего ее модуля.

# Виды алгоритмов.

При создании программ используются следующие алгоритмы:

- 1. Линейный алгоритм.*
- 2. Разветвляющийся алгоритм.*
- 3. Циклический алгоритм.*

# Условные операторы.

*Оператор if ... then ... else*

**if** условие **then**

операторы 1

**else**

операторы 2

**End if**

## *Оператор if ... elseif*

```
if условие 1 then  
    операторы 1  
elseif условие 2 then  
    операторы 2  
elseif условие 3 then  
    операторы 3  
...  
else  
    операторы n  
End if
```

# Операторы для создания циклов.

Оператор **For ... Next** используется для повторного выполнения операторов **заданное** число раз.

**For** счетчик = начало **To** конец [шаг]  
[Операторы]  
**Next** [счетчик]

Число выполнений этого оператора определяется параметрами **начало** и **конец**. Переменная **счетчик** при первом выполнении цикла имеет значение **начало** и каждый раз при выполнении оператора **Next** увеличивается на величину, заданную параметром **шаг**. Если параметр **шаг** отсутствует, то значение **счетчик** увеличивается на 1.

# Знакомство с массивами

Предположим, что у вас есть набор чисел (цены, оценки за экзамен и т.д.). Если эти числа расположить каждое в отдельной строке, т.е. в виде списка – это и есть **массив** чисел.

Каждый массив в VBA имеет **ИМЯ**, т.е. аналог заголовка списка на листе бумаги.

Чтобы работать с отдельным элементом массива, нужно сослаться на него **по имени и индексу** – *целому числу, соответствующему месту элемента в массиве.*

Массивы могут быть **многомерными**.

**Одномерный** массив – это вектор с координатами, а **двумерный** – это матрица.

Пример!!(Табулирование функции)

# Примеры

1. Создать функции пользователя для вычисления значений функций:

$$y = \frac{\sqrt[3]{x^2}}{(1-x)(1+x)^2}$$

$$y = \begin{cases} \cos^3 2x & x \leq -1 \\ e^x \sqrt[3]{x^2 + 1} & x > -1 \end{cases}$$