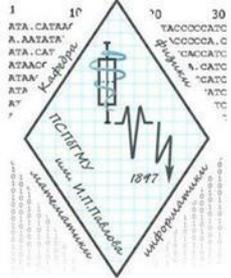


Язык программирования VBA

Тишков Артем Валерьевич

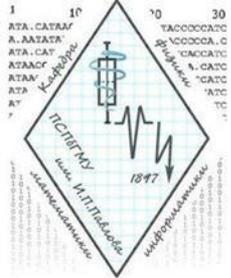
2018



1. Элементарное взаимодействие Excel и VBA

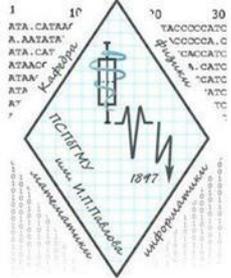
Рассматриваемые вопросы:

- Создание и выполнение макросов Excel
- Разработка пользовательской таблицы средствами процессора Excel
- Интегрированная среда разработки VBA
- Типы записи ссылок в Excel
- Анализ текста созданного макроса



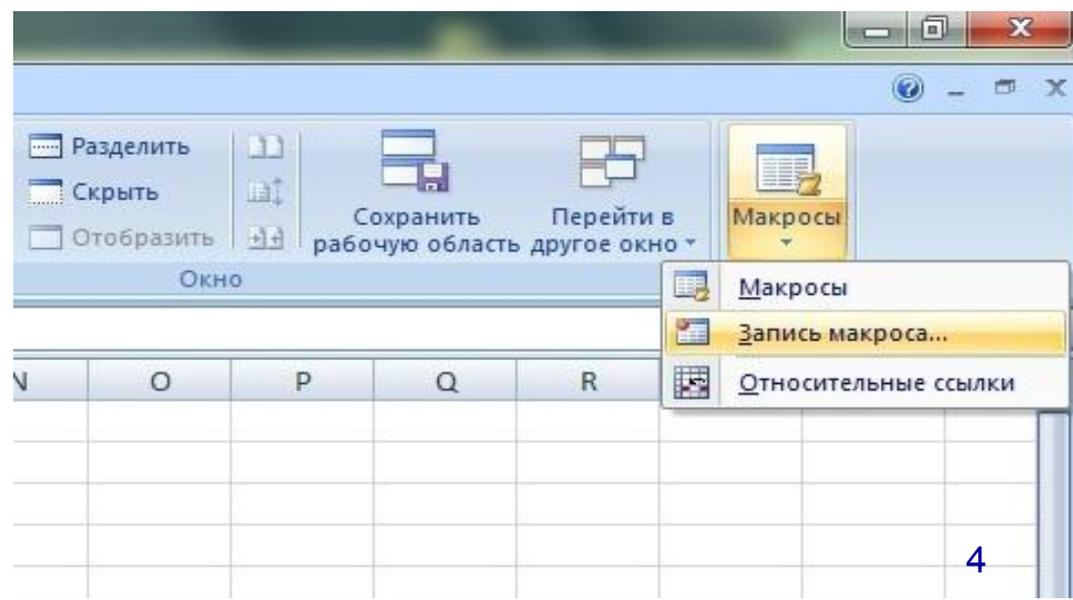
Создание и выполнение макросов Excel

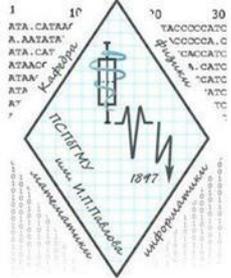
- *Макросом* обычно называют последовательность операторов на языке Visual Basic for Applications (VBA)
- Каждый макрос должен иметь собственное *имя*.
- По своей сути макрос представляет собой *программу* и может быть создан автоматически в специальном режиме работы программной системы (в том числе и *Excel*) или как результат программирования в терминах языка системы



Создание и выполнение макросов Excel

- Для создания макроса в Excel легче всего использовать автоматический режим его создания, вызываемый из главного меню Вид кнопкой **Макросы**.





Создание и выполнение макросов Excel

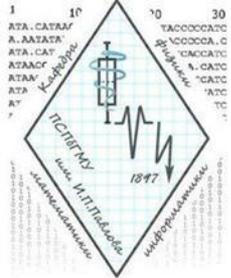
- Проверьте установку уровня безопасности командой **Параметры, Центр управления безопасностью, Параметры управления безопасностью, Параметры макросов.**

Параметры макросов

- Отключить все макросы без уведомления
- Отключить все макросы с уведомлением
- Отключить все макросы кроме макросов с цифровой подписью
- Включить все макросы (не рекомендуется, возможен запуск опасной программы)

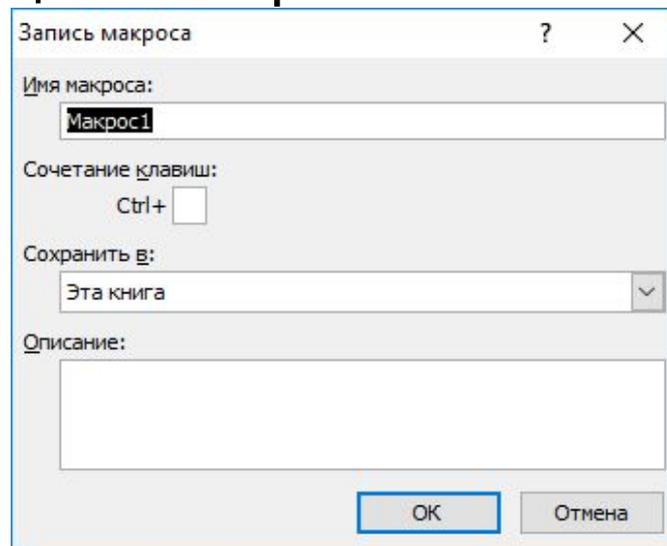
Параметры макросов для разработчика

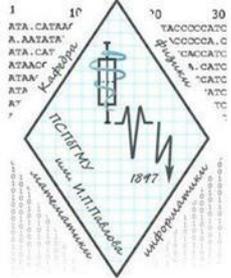
- Доверять доступ к объектной модели проектов VBA



Создание и выполнение макросов Excel

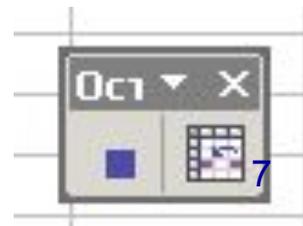
- Если в меню **Вид, Макрос** выбрать пункт **Запись макроса...**, то откроется диалоговое окно, позволяющее задать имя макроса и, при желании, комбинацию клавиш, с помощью которой он также может вызван

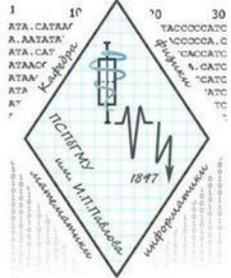




Создание и выполнение макросов Excel

- Начиная с этого момента все действия с рабочей книгой дополнительно записываются в файл макроса
- Остановить запись макроса можно кнопкой **Остановить запись** дополнительно открывшейся панели инструментов или через аналогичный пункт главное меню **Вид, Макрос**.

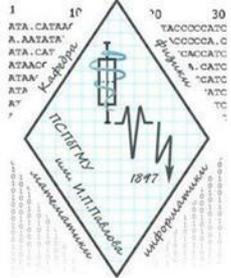




Разработка пользовательской таблицы средствами процессора Excel

	A	B	C	D	E
1	Фамилия, И. О.	Начислено	Налог	К выдаче	
2	Иванов В. Н.	1 234,00р.			
3	Трофимова Л. А.	1 234,00р.			
4	Семенова Е. Г.	1 000,00р.			
5	Стапанов А. Г.	900,00р.			
6	Итого	4 368,00р.			
7	Ставка подоходного налога		12,00%		
8					

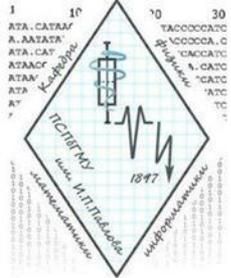
Исходные данные



Разработка пользовательской таблицы средствами процессора Excel

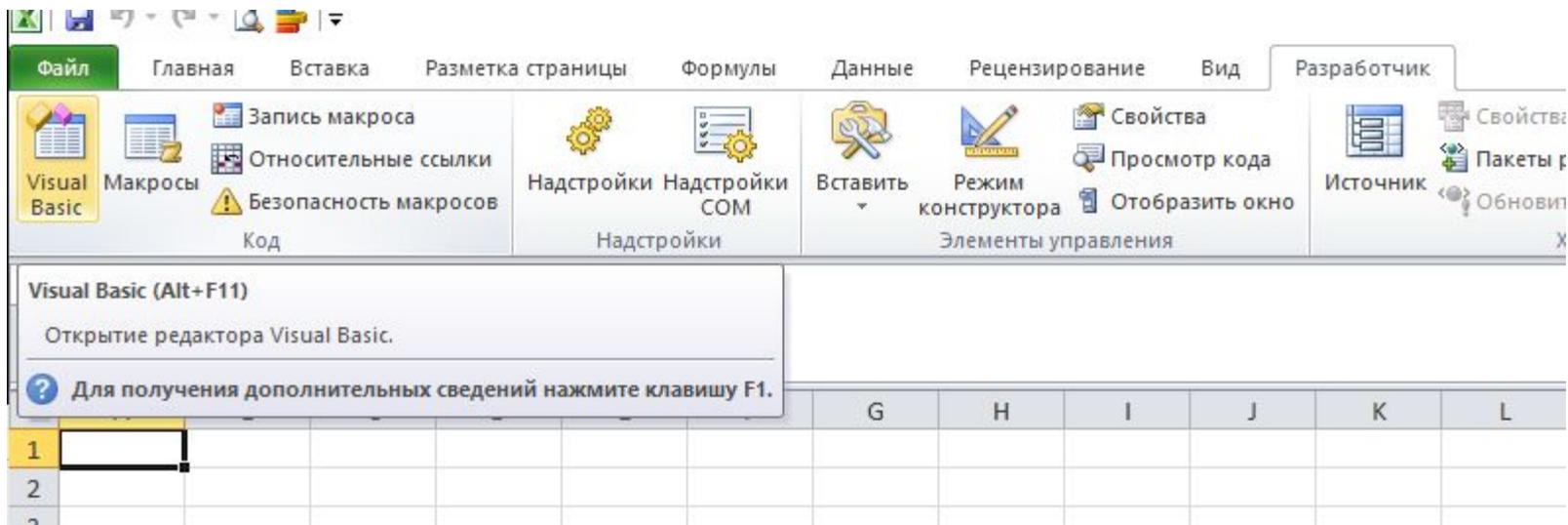
	A	B	C	D	E
1	Фамилия, И. О.	Начислено	Налог	К выдаче	
2	Иванов В. Н.	1 234,00р.	148,08р.	1 085,92р.	
3	Трофимова Л. А.	1 234,00р.	148,08р.	1 085,92р.	
4	Семенова Е. Г.	1 000,00р.	120,00р.	880,00р.	
5	Стапанов А. Г.	900,00р.	108,00р.	792,00р.	
6	Итого	4 368,00р.	524,16р.	3 843,84р.	
7	Ставка подоходного налога		12,00%		
8					

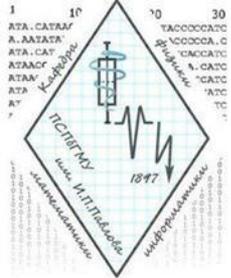
Результаты программирования в Excel



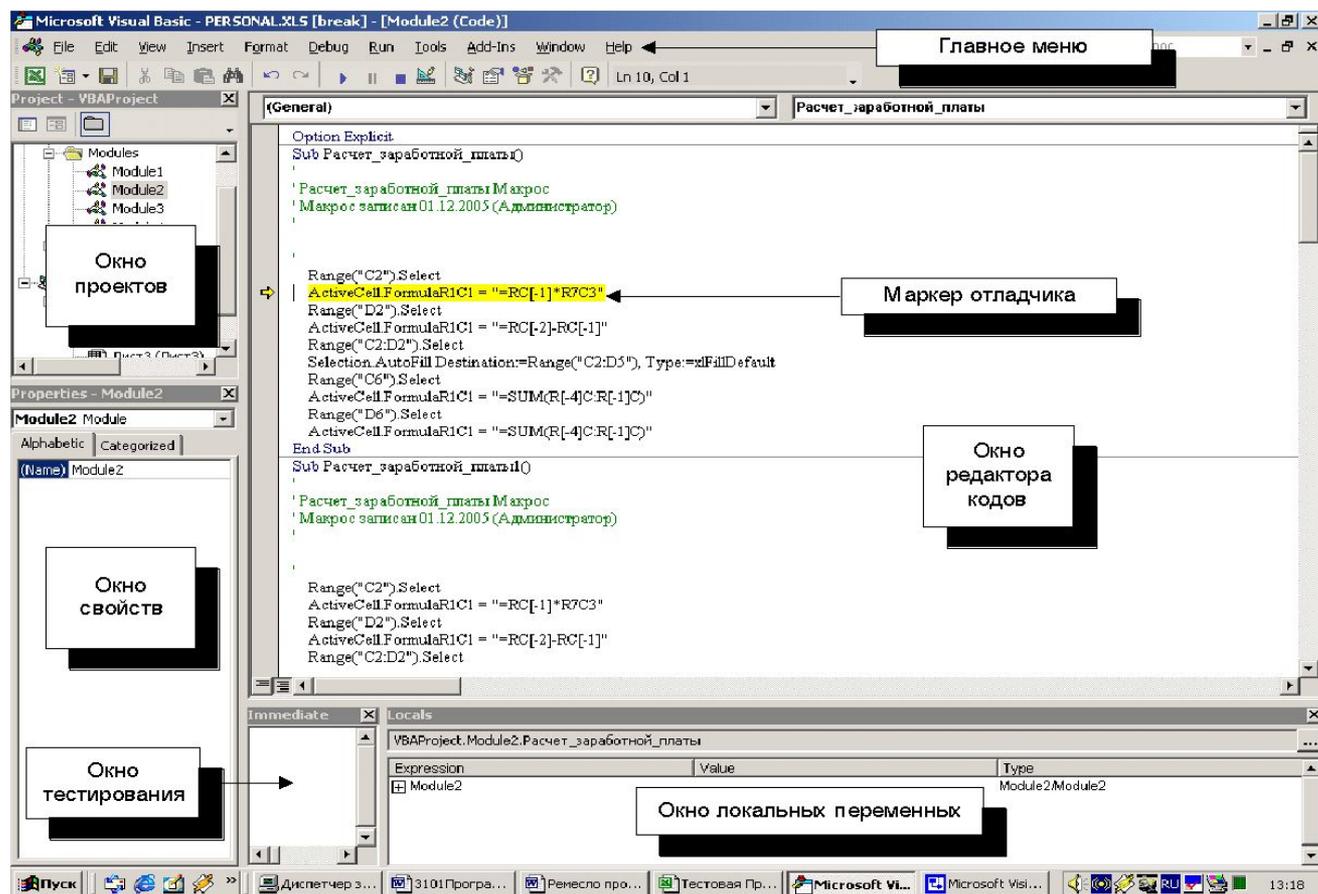
1.3. Интегрированная среда разработки VBA

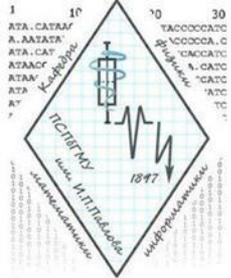
- Запуск редактора VBA





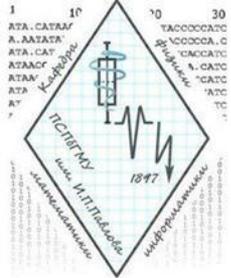
Интегрированная среда разработки VBA





1.4. Типы записи ссылок в Excel

- В Excel используются два типа записи ссылок на ячейки в формулах: A1 (по умолчанию) и R1C1.
- A1: первый символ имя столбца, второй – номер строки
- R1C1: первые два символа “Row” + “1” – номер строки, вторые - “Column” + “1” – номер столбца. В программировании используется этот тип записи ссылок.

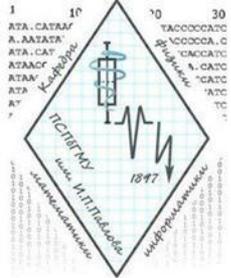


Типы записи ссылок в Excel

- Использование этого стиля позволяют организовать относительную и абсолютную адресацию к ячейкам таблицы (за счет введения в строку символа \$).

Относительный адрес в формате A1: B3

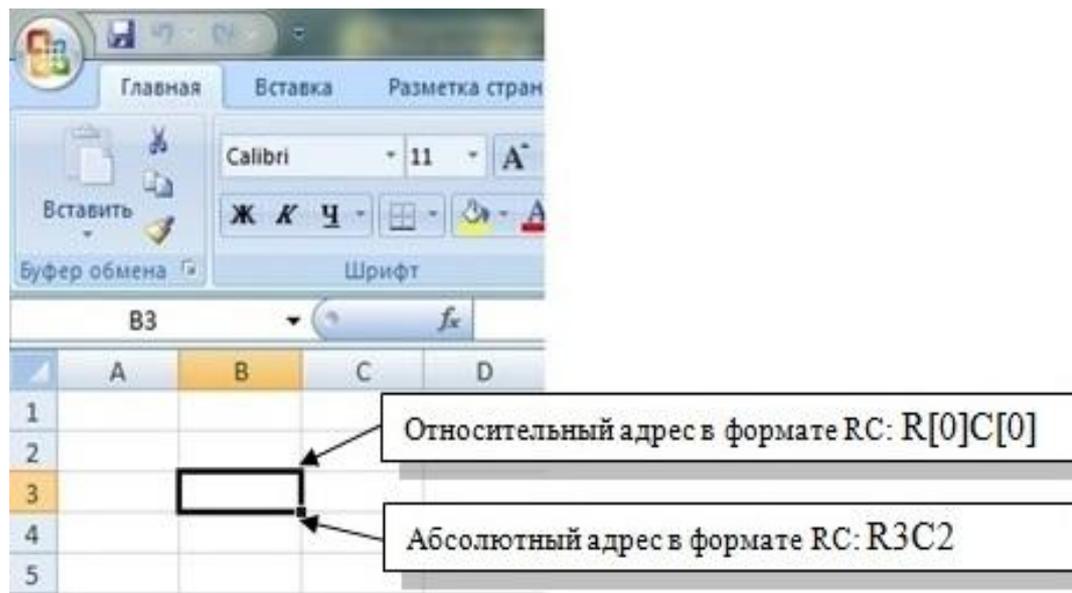
Абсолютный адрес в формате A1: \$B\$3

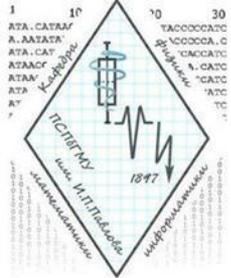


Типы записи ссылок в Excel

- При использовании абсолютной адресации после символов R и C указывается собственно номер строки и столбца. Так, например, ячейка \$B\$3 имеет адрес R3C2.

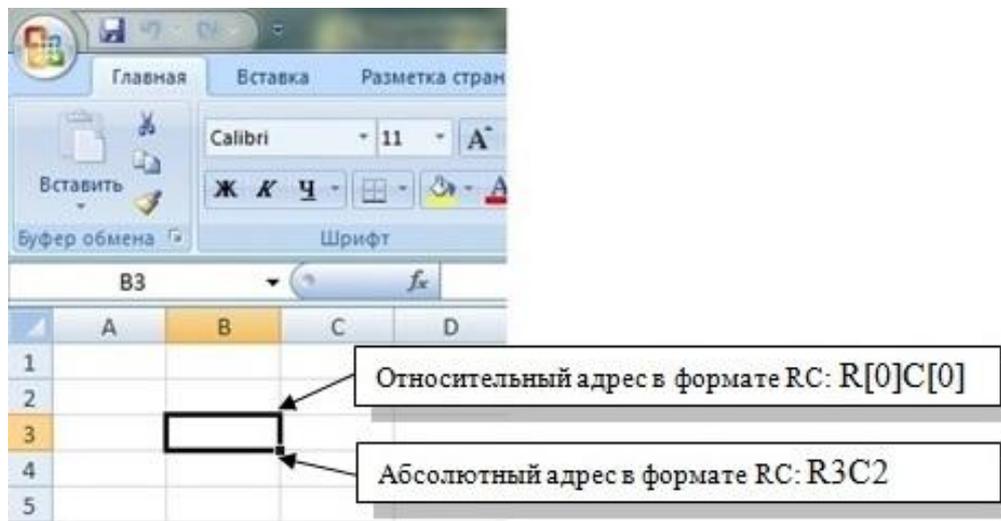
Обратите внимание на то, что, в отличие от типа A1, при использовании типа ссылок R1C1 сначала записывается строка, а потом столбец.

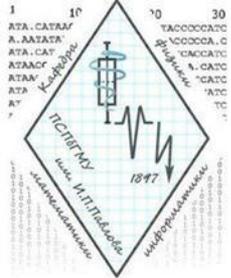




Типы записи ссылок в Excel

- При использовании относительной адресации в стиле R1C1 после обозначения строки или колонки в квадратных скобках указывается смещение по отношению к текущей ячейке.





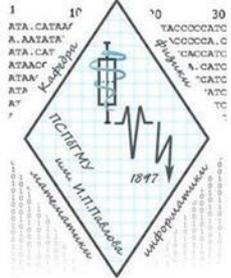
Типы записи ссылок в Excel

- Так, например, если данные находятся в ячейке В3, а ссылка на нее программируется в ячейке А5, то в формуле она запишется как R[-2]C[1].

Эта запись может интерпретироваться как обращение к ячейке, находящейся на одну строку выше и две колонки правее текущей.

Относительный адрес данных в формате RC: R[-2]C[1]

Абсолютный адрес данных в формате RC: R3C2



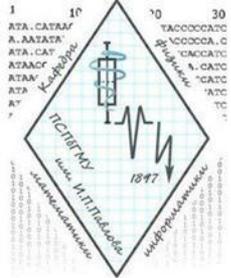
1.5. Анализ текста созданного макроса

Sub Расчет_заработной_платы()
 Расчет_заработной_платы Макрос
 Макрос записан 01.12.2005 (Администратор)

```

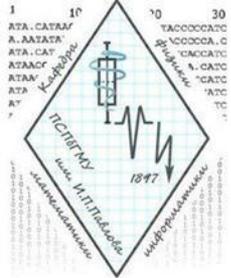
Range("C2").Select
ActiveCell.FormulaR1C1 = "=RC[-1]*R7C3"
Range("D2").Select
ActiveCell.FormulaR1C1 = "=RC[-2]-RC[-1]"
Range("C2:D2").Select
Selection.AutoFill Destination:=Range("C2:D5"),
Type:=xlFillDefault
Range("C6").Select
ActiveCell.FormulaR1C1 = "=SUM(R[-4]C:R[-1]C)"
Range("D6").Select
ActiveCell.FormulaR1C1 = "=SUM(R[-4]C:R[-1]C)"
End Sub
  
```

	A	B	C	D
1	Фамилия, И.О.	Начислено	Налог	К выдаче
2	Иванов В.Н.	1 234,00р.	148,08р.	1 085,92р.
3	Трофимова Л.А.	1 234,00р.	148,08р.	1 085,92р.
4	Семенова Е.Г.	1 000,00р.	120,00р.	880,00р.
5	Степанов А.Г.	900,00р.	108,00р.	792,00р.
6	Итого	4 368,00р.	524,16р.	3 843,84р.
7	Ставка подоходного налога		12,00%	



Анализ текста созданного макроса

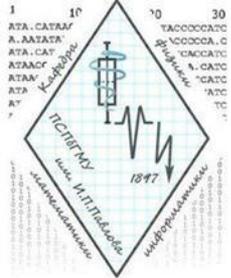
- Range (диапазон). Возникает при выделении
- ActiveCell (активная ячейка). Возвращает объект Range
- FormulaR1C1. Свойство, возвращающее или задающее формулу типа R1C1 в активную ячейку
- Selection (выделение). Свойство, возвращающее выделенный объект
- AutoFill (автозаполнение). Метод, осуществляющий заполнение выделенных ячеек



2. Отладка и выполнение программы в среде VBA

Рассматриваемые вопросы:

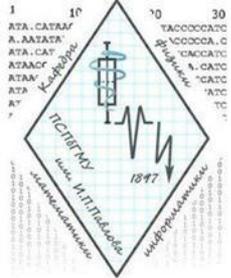
- Назначение окон интегрированной среды разработки VBA
- Выполнение программы в автоматическом режиме
- Выполнение программы в режиме отладки



2.1. Назначение окон интегрированной среды разработки VBA

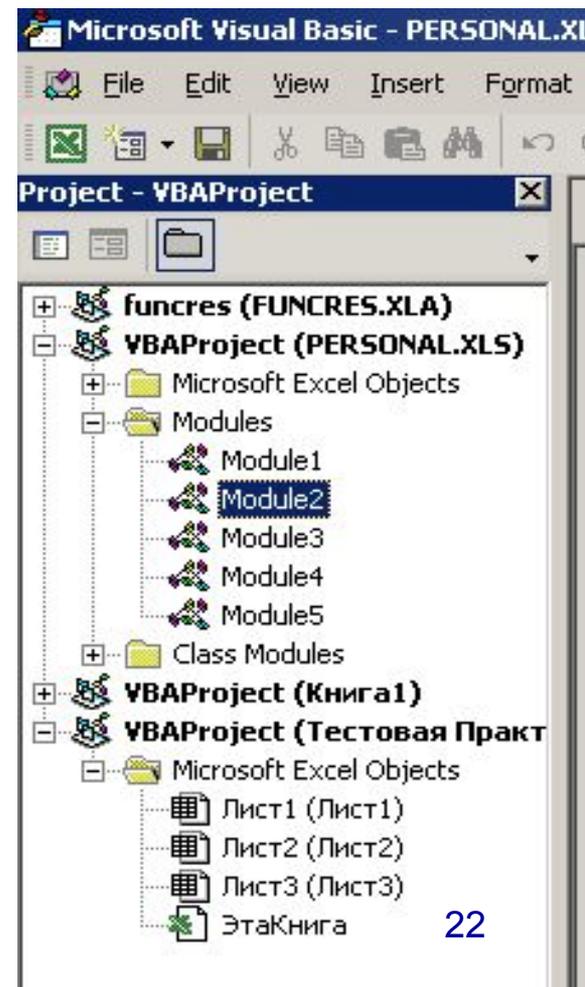
The screenshot shows the Microsoft Visual Basic IDE with the following components and callouts:

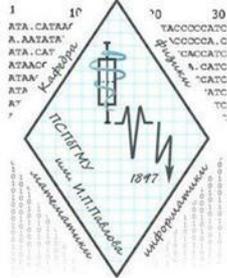
- Project Explorer (Окно проектов):** Shows a tree view of the project structure with Modules 1, 2, and 3.
- Properties Window (Окно свойств):** Shows the properties for the selected Module2.
- Code Editor (Окно редактора кодов):** Contains VBA code for the macro 'Расчет_заработной_платы'. A yellow highlight and a callout 'Маркер отладчика' (Breakpoint Marker) are placed on the line: `ActiveCell.FormulaR1C1 = "=RC[-1]*R7C3"`.
- Test Window (Окно тестирования):** Located at the bottom left, used for testing the macro.
- Locals Window (Окно локальных переменных):** Located at the bottom right, showing the current state of local variables.



Назначение окон интегрированной среды разработки VBA

- Окно *проектов* содержит список форм и модулей текущего проекта.
- *Проект* – набор файлов, используемых для построения приложений.

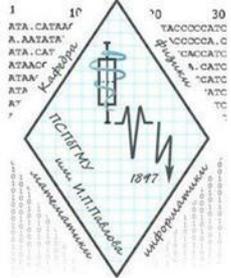




Назначение окон интегрированной среды разработки VBA

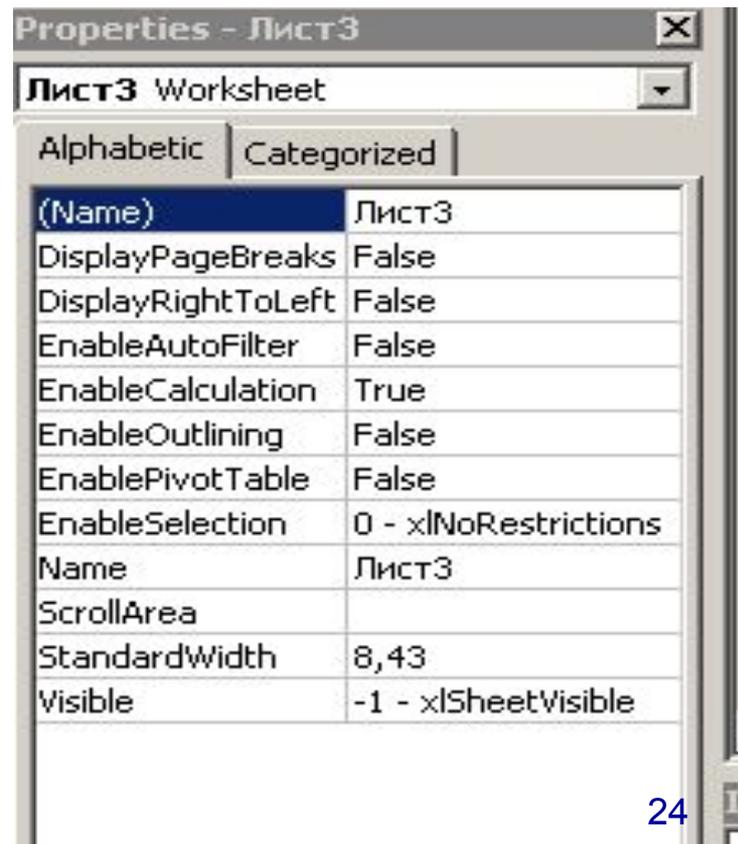
- Окно *редактора кодов* служит для редактирования программного кода приложения. Для каждой формы и каждого модуля кода создается свое окно.

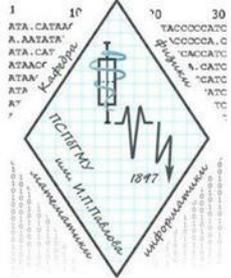
```
Sub Макрос1()  
PERSONAL.XLS - Module2 (Code)  
(General) | Расчет_заработной_платы  
Option Explicit  
Sub Расчет_заработной_платы()  
' Расчет_заработной_платы Макрос  
' Макрос записан 01.12.2005 (Администратор)  
  
Range("C2").Select  
ActiveCell.FormulaR1C1 = "=RC[-1]*R7C3"  
Range("D2").Select  
ActiveCell.FormulaR1C1 = "=RC[-2]-RC[-1]"  
Range("C2:D2").Select  
Selection.AutoFill Destination:=Range("C2:D5"), Type:=xlFillDefault  
Range("C6").Select  
ActiveCell.FormulaR1C1 = "=SUM(R[-4]C:R[-1]C)"  
Range("D6").Select  
ActiveCell.FormulaR1C1 = "=SUM(R[-4]C:R[-1]C)"  
End Sub
```



Назначение окон интегрированной среды разработки VBA

- Окно *свойств* перечисляет установленные свойства выбранного объекта

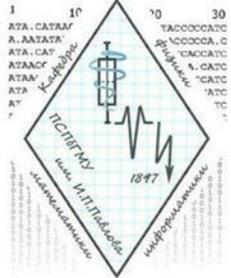




Назначение окон интегрированной среды разработки VBA

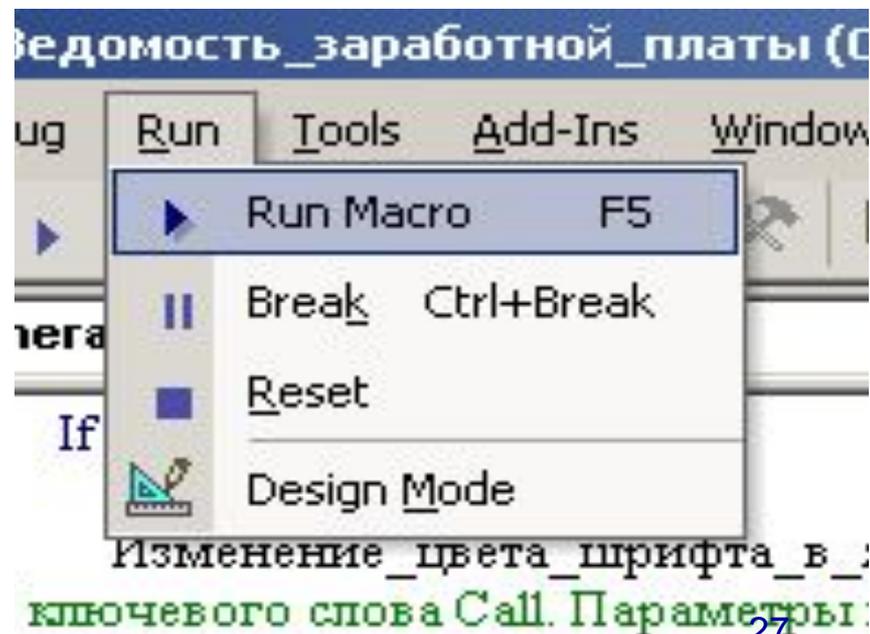
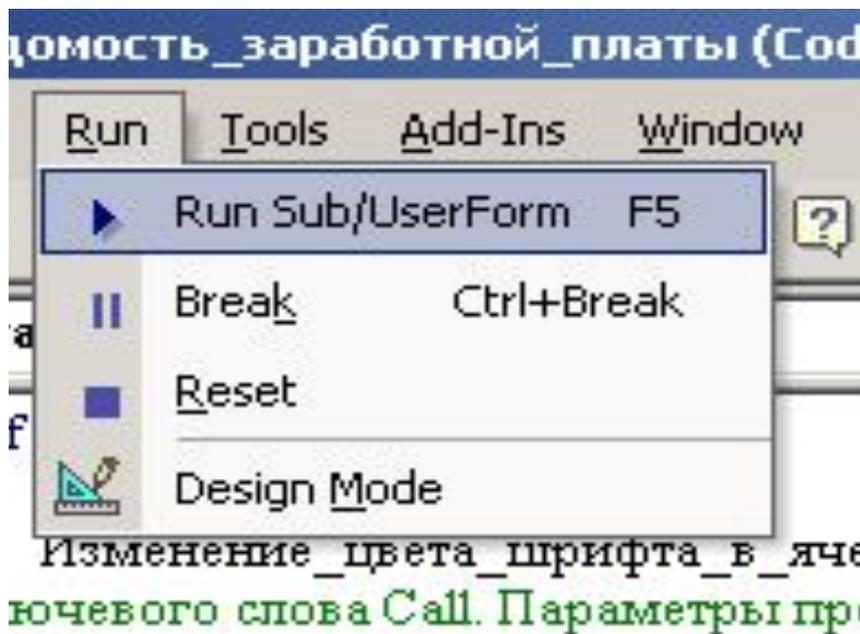
Кроме перечисленных интегрированная среда разработки содержит окна

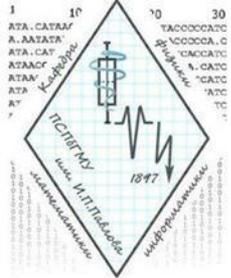
- *тестирования Immediate* (немедленное выполнение), позволяющее изменять значения переменных программы в момент ее выполнения и даже вводить дополнительные операторы;
- *просмотра мгновенных значений Watch*, позволяющее вести контроль выбранной переменной программы;
- некоторые другие.



2.2. Выполнение программы в автоматическом режиме

- Run, Run Sub/User Form
- Run, Run Macro

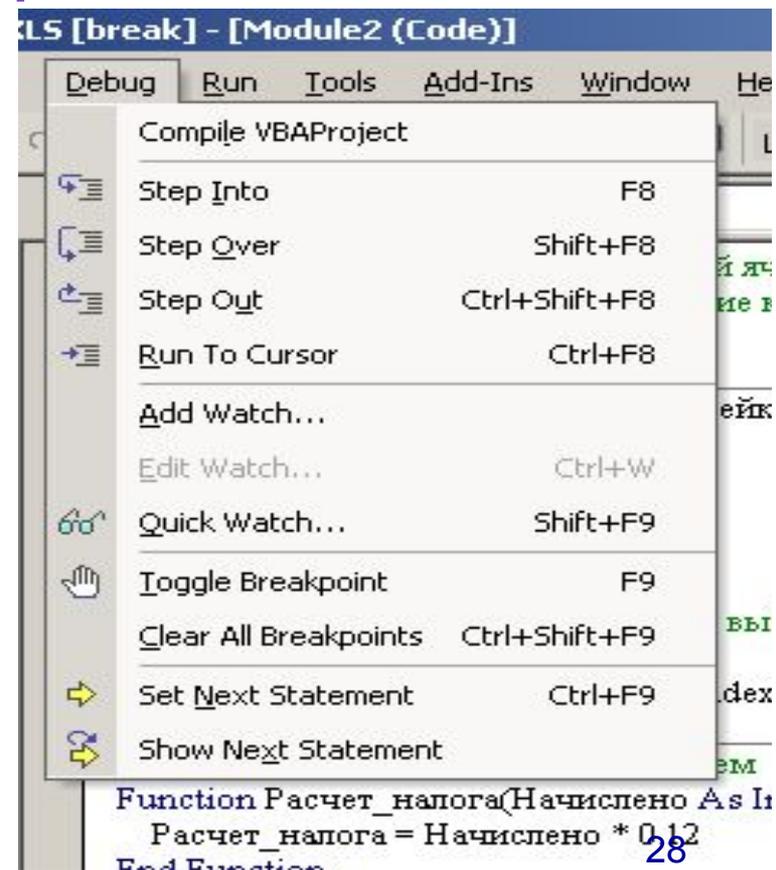


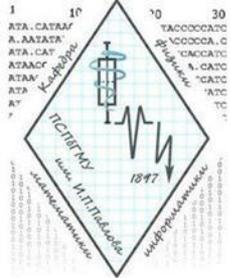


2.3. Выполнение программы в режиме отладки

- **Debug, Step Into**
- **Debug, Step Over**
- **Debug, Step Out**
- **Debug, Run To Cursor**

- **Debug, Toggle Breakpoint**

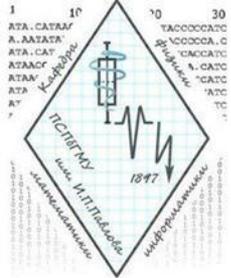




3. Обмен данными между Excel и VBA

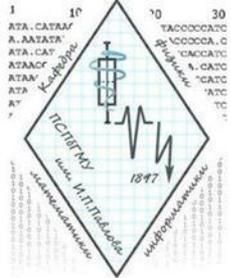
Рассматриваемые вопросы:

- Типы данных VBA
- Идентификаторы в VBA
- Объявление переменных в VBA
- Массивы в VBA
- Оператор присваивания
- Использование свойства Cells() для обмена данными между Excel и VBA

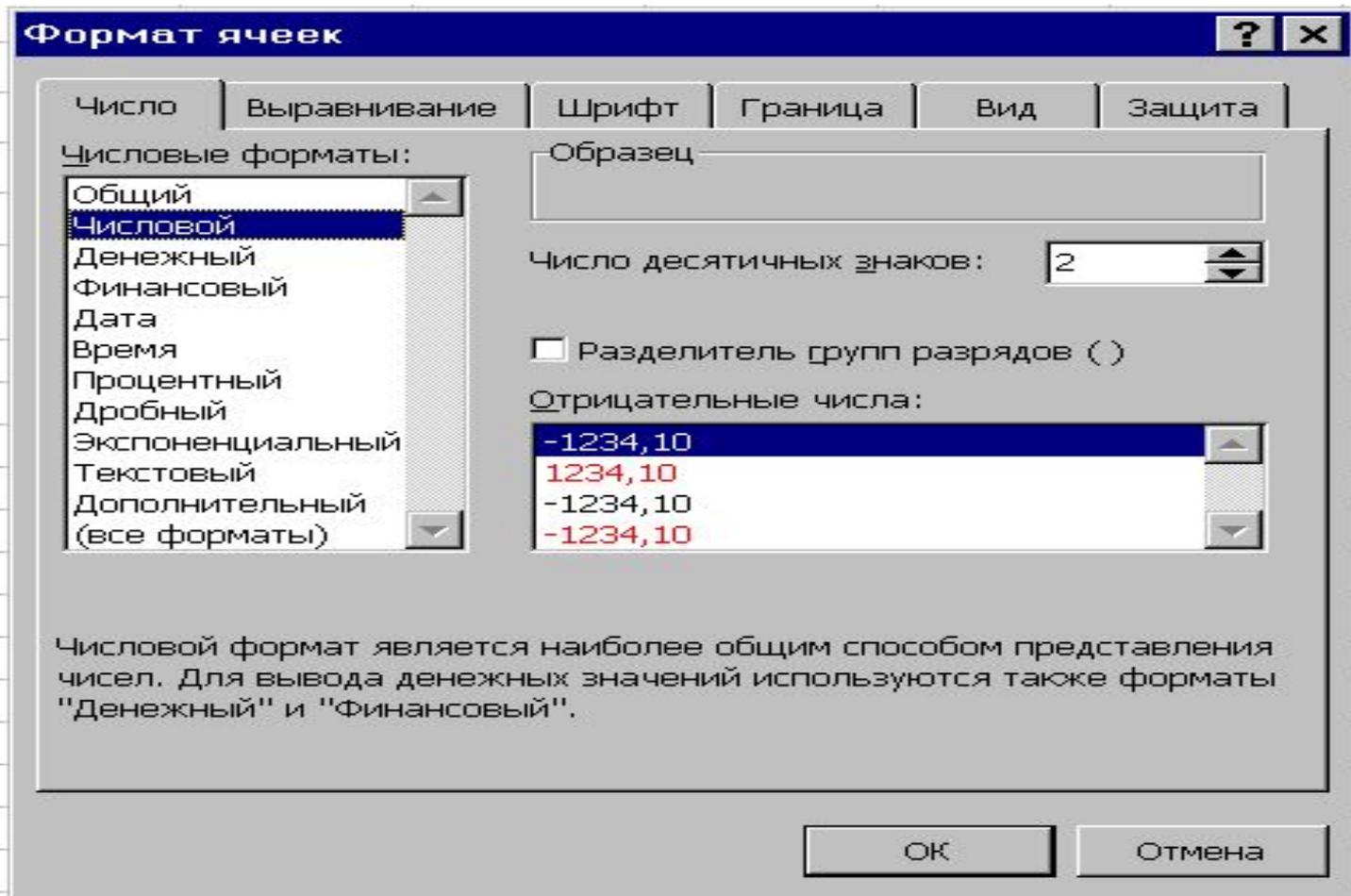


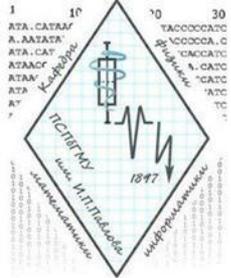
3.1. Типы данных VBA

- Тип данных - способ внутреннего представления данных в памяти машины, учитывающий метод их кодирования в одной или нескольких ячейках памяти и предусматривающий возможности их расшифровки или преобразования.



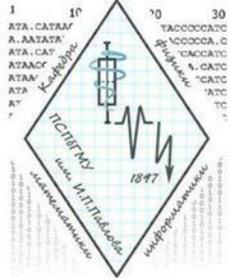
Типы данных VBA





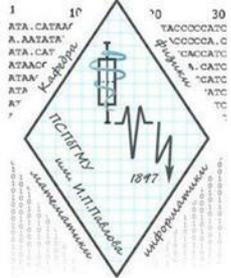
Типы данных VBA

Тип данных	байт		Диапазон значений
Byte (байт)	1		От 0 до 255
Boolean (логический)	2		True или False
Integer (целые)	2	%	От -32768 до 32767
Long (длинное целое)	4	&	От -2147483648 до 2147483647
Single (плавающее обычной точности)	4	!	От -3,402823E38 до -1,401298E-45 и от 1,401298E-45 до 3,402823E38
Double (плавающее двойной точности)	8	#	От -1,79769313486231E308 до -4,94065645841247E-324 и от 4,94065645841247E-324 до 1,79769313486231E308
Currency (денежный)	8	@	От -922337203685477,5808 до 922337203685477,5808
Decimal (масштабируемое целое)	14		+/-7922816251426433759353950335 и 28 знаков после запятой. Минимальное отличное от нуля значение имеет вид +/-0,00000000000000000000000000000001



Типы данных VBA

Тип данных	байт		Диапазон значений
Date (время и дата)	8		От 1 января 100 г. до 31 декабря 9999 г.
Object (объект)	4		Любой указатель объекта
String (строка переменной длины)	10+ длина строки		От 0 до приблизительно двух миллиардов
String (строка постоянной длины)	Длина строки	\$	От 1 до 65400
Variant (числовые подтипы)	16		От -1,79769313486232E308 до -4,94065645841247E-324 и от 4,94065645841247E-324 до 1,79769313486232E308
Variant (строковые подтипы)	22+ длина строки		От 0 до приблизительно двух миллиардов
Type (определяемый пользователем)	Определяется элементами типа		Диапазон каждого элемента определяется его типом данных



Типы данных VBA

- Type (определяемый пользователем).
Определяется элементами типа. Диапазон каждого элемента определяется его типом данных

Type Запись_Ведомости

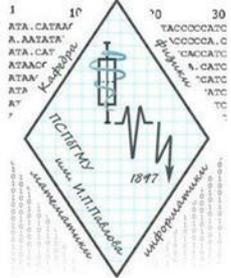
Фамилия_И_О As String

Начислено_Ведомость As Currency

Налог_Ведомость As Currency

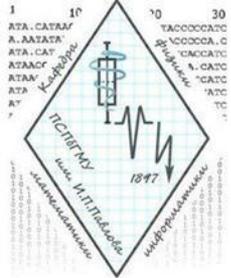
К_выдаче_Ведомость As Currency

End Type



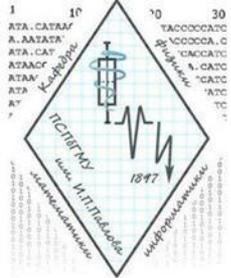
3.2. Идентификаторы в VBA

- Идентификатором называется символическое имя ячейки памяти.
- Каждый язык программирования содержит свои правила составления таких имен.
- Общим является то, что программист вправе сам придумать имя, что позволяет ему сохранить в нем смысловое значение.



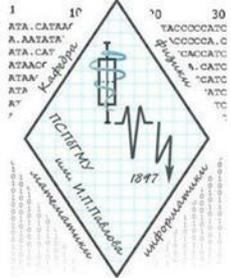
Идентификаторы в VBA

-
-
-
-
-
-



Идентификаторы в VBA

- Использование символов кириллицы в именах позволяет программисту создавать осмысленные имена идентификаторов, что облегчает чтение и отладку программы



Идентификаторы в VBA

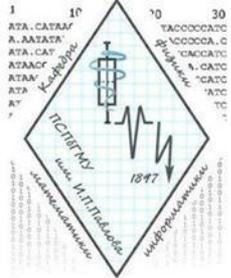
Возможные варианты идентификаторов языка VBA:

- `I`, `j`, `Name`, `Переменная`, `Результат_вычислений`.

Еще варианты записи идентификаторов:

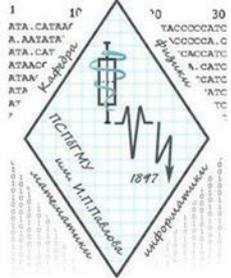
- `A%`, `B&`, `C!`, `D#`, `E@`, `F$`.

В этом случае символы `%`, `&`, `!`, `#`, `@`, `$` не входят в состав идентификатора и используются в качестве специального признака типа данных



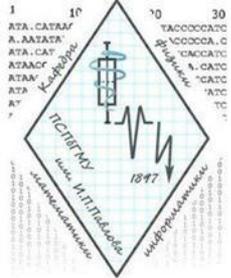
3.3. Объявление переменных в VBA

- Dim I As Integer, Name, j As Integer, Переменная As Integer, GGG As Integer
- Обратите внимание на то, что если вы не указываете явно тип переменной, то по умолчанию она имеет тип **Variant**. Так, в рассмотренном выше примере такой тип имеет переменная **Name**.



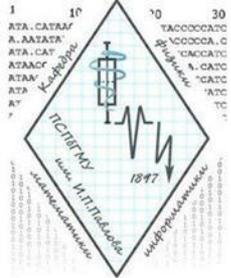
Объявление переменных в VBA

- Обработывая файл исходного текста программы компилятор создает двоичный файл, который впоследствии после дополнительной обработки представляет собой последовательность кодов программы, выполняемой процессором
- Там же в программе отводится область для хранения данных



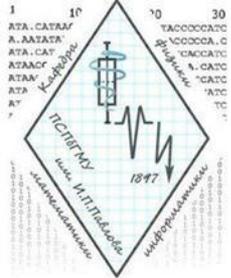
Объявление переменных в VBA

- Строка объявления переменных обрабатывается компилятором. Результат обработки – резервирование ячеек памяти в двоичном файле
- Адреса ячеек определяются как смещение по отношению к первому байту исполняемой программы
- После загрузки файла в ОЗУ адреса ячеек получают абсолютные значения



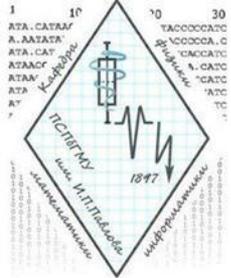
3.4. Массивы в VBA

- Практика программирования широко использует переменные, обращение к которым ведется как по имени, так и по номеру.
- В этом случае можно говорить о создании переменных табличного типа, когда обращение к данным ведется по имени и номеру (индексу) внутри этого имени.
- Такие переменные обычно называются массивами.



Массивы в VBA

- Массив - последовательно упорядоченные в памяти данные одного типа
- Каждый массив имеет имя
- Имя массива – идентификатор, за которым закреплен свой адрес ОЗУ



Массивы в VBA

- Количество таких ячеек определяет размер массива
- Объявления массивов:

`Dim YY(25)`

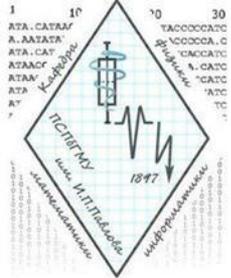
Объявляется одномерный массив из 26 элементов.

Начальный (базовый) индекс принят по умолчанию равным 0.

`Dim ZZ(3,10) As Single`

Объявляется двумерный массив ZZ типа Single, первый индекс которого меняется в диапазоне от 0 до 3, а второй в диапазоне от 0 до 10.

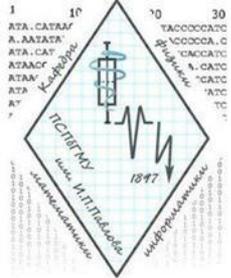
`Dim SS(-3 To 3,1 To 10) As Integer`



Массивы в VBA

Dim SS(-3 To 3,1 To 10) As Integer

- Обращение к элементу массива в тексте программы с явным указанием номеров элементов: SS(-2,5).
- Если переменная Name содержит число -2, а ячейка Переменная число 5, то обращение SS(Name, Переменная) полностью эквивалентно предыдущему.
- Если в процессе предыдущих вычислений переменная Name примет значение -4, а мы попытаемся выполнить SS(Name, Переменная), то произойдет обращение к несуществующему элементу массива и возникнет ошибка выхода индекса за границы массива.



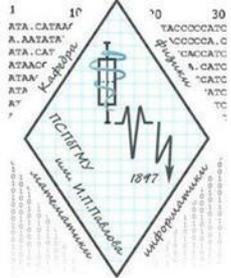
Динамические массивы в VBA

Dim Начислено() As Currency, i As Integer

i = 10

ReDim Начислено(1 To i)

- Массив Начислено() первоначально был объявлен как массив неопределенной длины. Инструкция ReDim изменила массив, причем память под него была отведена в момент выполнения программы.

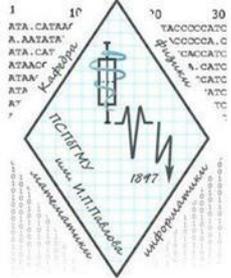


3.5. Оператор присваивания

- Оператор присваивания обеспечивает занесение информации в ячейки памяти, связанные с идентификатором и имеет символ равенства (=).

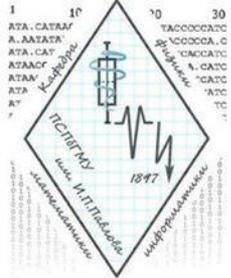
$$i = 10$$

- В отличие обычного равенства, которое выполняется всегда, оператор присваивания имеет динамические свойства (зависит от времени).



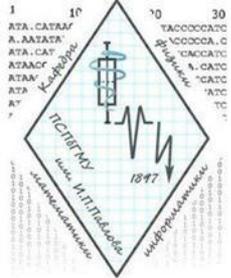
3.6. Использование свойства Cells() для обмена данными между Excel и VBA

- Отдельную проблему представляет прямая и обратная передача данных из таблицы Excel в ячейки памяти, объявленные в программе, написанной на VBA.
- Автоматически созданный макрос непосредственно манипулирует с ячейками таблицы используя стили ссылки на ячейки в Excel: A1 и R1C1.
- Такой прием может быть использован и в рабочей программе, однако в этом случае ее модификация и использование существенно затруднены.



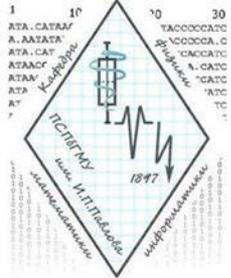
Использование свойства Cells() для обмена данными между Excel и VBA

- Гораздо предпочтительнее использовать свойство Cells() стандартного объекта Excel Range.
- Сам объект представляет собой ячейку, столбец, строку или выделенный диапазон листа Excel.
- Свойство Cells() позволяет непосредственно обратиться к объекту Excel по номеру строки и колонки.
- Поскольку это свойство установлено по умолчанию для рабочего листа Excel, то его можно использовать без дополнительных указаний.



Использование свойства Cells() для обмена данными между Excel и VBA

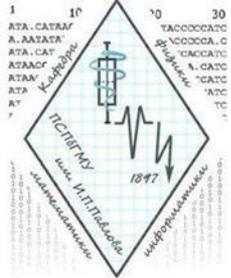
- Если запись свойства стоит слева от оператора присваивания, то производится запись данных в ячейку таблицы, если справа, то считывание значения из ячейки таблицы.
- Кроме собственно записи данных свойство Cells() в сочетании со свойствами других объектов (Font, Color и т.п.) позволяет задавать параметры шрифта, его цвет, фон и так далее.



4. Операции VBA

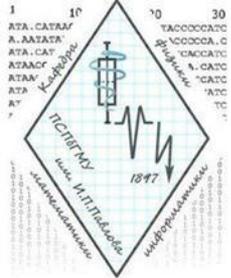
Рассматриваемые вопросы:

- Арифметические операции
- Операции сравнения
- Логические операции
- Операции со строками



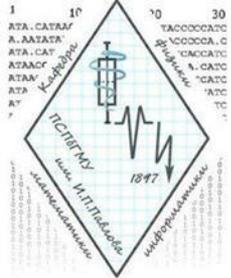
4.1. Арифметические операции

Арифметические операции	Приоритет	Название	$A=11$ $B=5$	Результат
-	3	Смена знака	-A	-11
+	7	Сложение	A+B	16
-	7	Вычитание	A-B	6
*	4	Умножение	A*B	55
/	4	Деление	A/B	2.2
\	5	Целочисленное деление	A\B	2
Mod	6	Остаток от деления по модулю	A Mod B	1
^	2	Возведение в степень	A^B	161015



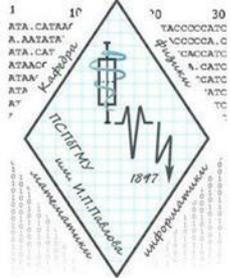
4.2. Операции сравнения

Операции	Приорите	Название	Пример	Результат
Операции сравнения			A=11 B=5	
<	8	Меньше	A<B	False
>	8	Больше	A>B	True
<=	8	Меньше и равно	A<=B	False
>=	8	Больше и равно	A>=B	True
<>	8	Не равно	A<>B	True
=	8	Равно	A=B	False
Is		Сравнение со ссылкой на объекты	Dim A,B,C,D,E Set A=D Set B=D Set C=E F=A Is B E=A Is C	True False



4.3. Логические операции

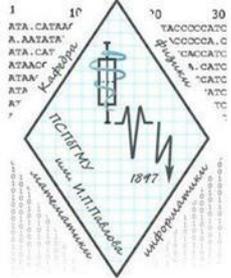
Операции	Приоритет	Название	Пример	Результат
Логические операции			A=True B=True C=False D=False	
And	10	Логическое умножение (и)	A And B A And C C And B C And D	True False False False
Or	11	Логическое сложение (или)	A Or B A Or C C Or B C Or D	True True True False
Xor	12	Исключающее или	A Xor B A Xor C C Xor B C Xor D	False True True False
Not	9	Отрицание	E=Not B E=Not D	False True
Imp	14	Импликация	A Imp B A Imp C C Imp B C Imp D	True False True True
Eqv	13	Эквивалентность	A Eqv B A Eqv C C Eqv B C Eqv D	True False False True



5. Операторы VBA

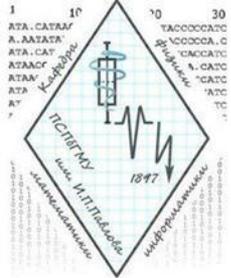
Рассматриваемые вопросы:

- Правила записи операторов в языке VBA
- Условный оператор
- Оператор ветвления
- Семейство операторов For
- Семейство операторов Do



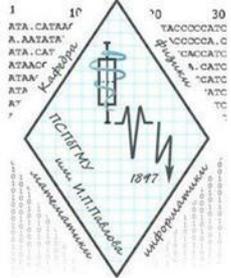
Операторы VBA

- Оператором называется самостоятельная конструкция языка программирования, которая может быть отдельно откомпилирована и выполнена в виде заранее определенной последовательности кодов процессора



5.1. Правила записи операторов в языке VBA

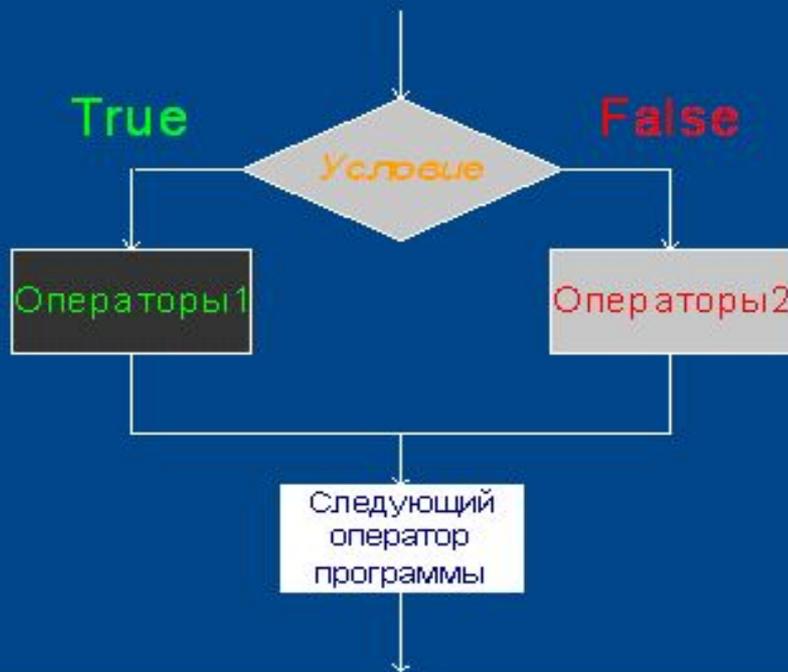
- Операторы записываются на отдельных строках и могут не нумероваться.
- Для размещения нескольких операторов на одной строке между ними необходимо поставить символ двоеточие (:). Этот же символ используется для обозначения меток.
- Для переноса продолжения оператора на следующую строку используется комбинация символов пробел знак подчеркивания (_). Нельзя разбивать переносом выражения и строки. Допускается не более семи переносов строк одного оператора.

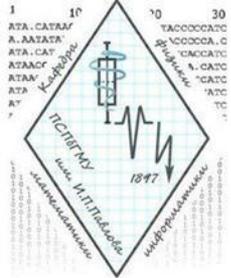


5.2. Условный оператор



If **Условие** Then **Операторы1** Else: **Операторы2** Endif Следующий оператор программы





Условный оператор

- Формат условного оператора

If Условие Then [Операторы] [Else
Операторы_Else] End If

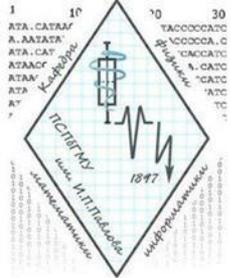
Пример программы с условным оператором

If Таблица(i) > extr Then

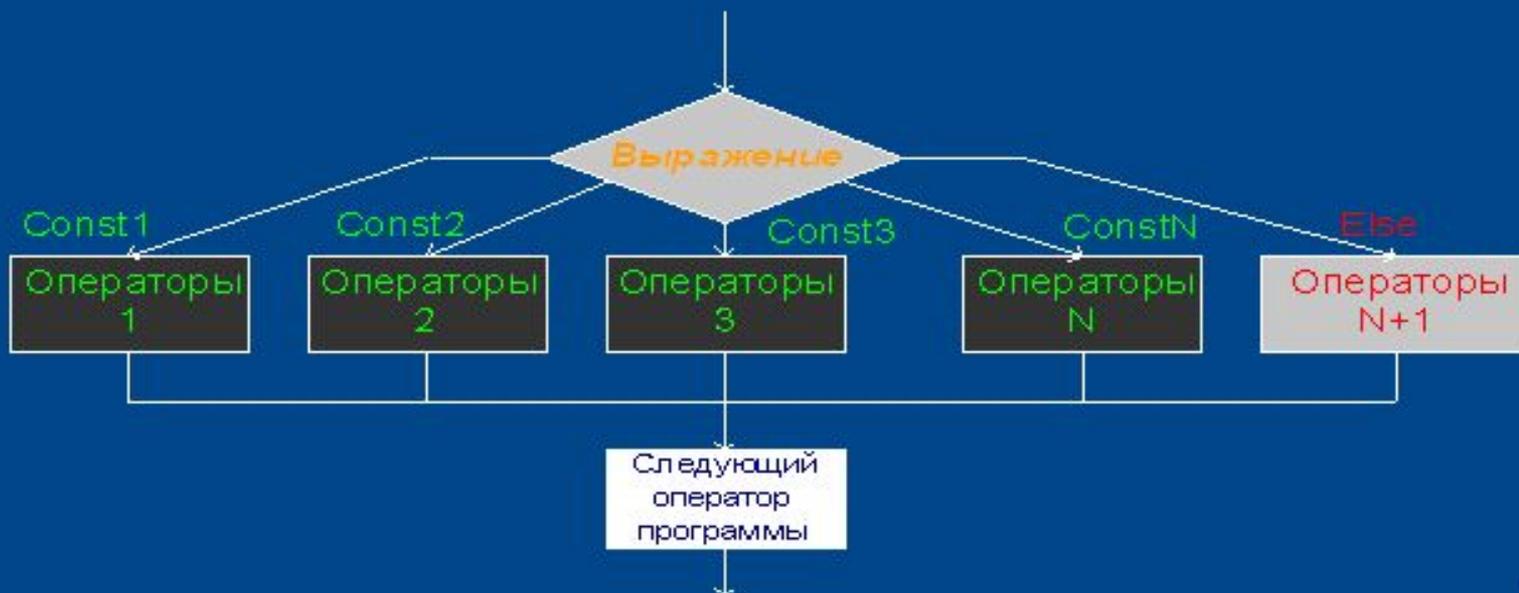
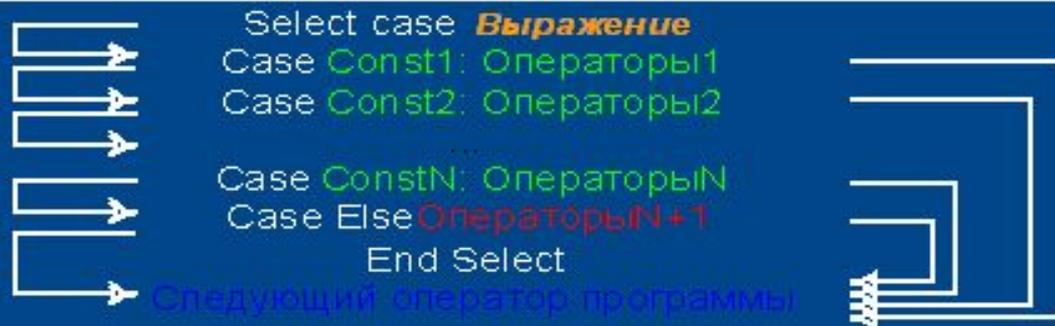
extr = Таблица(i)

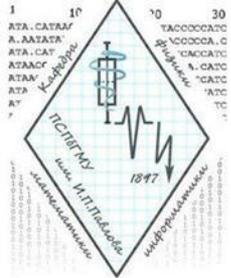
Else

End If



5.3. Оператор ветвления





Оператор ветвления

- Формат оператора ветвления

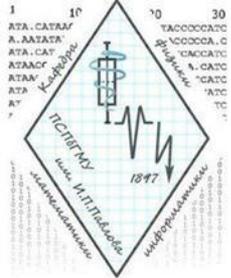
Select Case Выражение

[Case Значение1 [Операторы1]]

[Case ЗначениеN [ОператорыN]]

[Case Else [ОператорыElse]]

End Select



Оператор ветвления

- Пример программы с оператором ветвления

```
Dim РежимРаботы As String, День As Integer
```

```
День = 2
```

```
Select Case День
```

```
Case 1
```

```
РежимРаботы = "Прием документов"
```

```
Case 2, 3, 4
```

```
РежимРаботы = "Работа с документами"
```

```
Case 5
```

```
РежимРаботы = "Выдача документов"
```

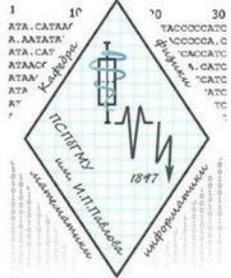
```
Case 6, 7
```

```
РежимРаботы = "Выходные дни"
```

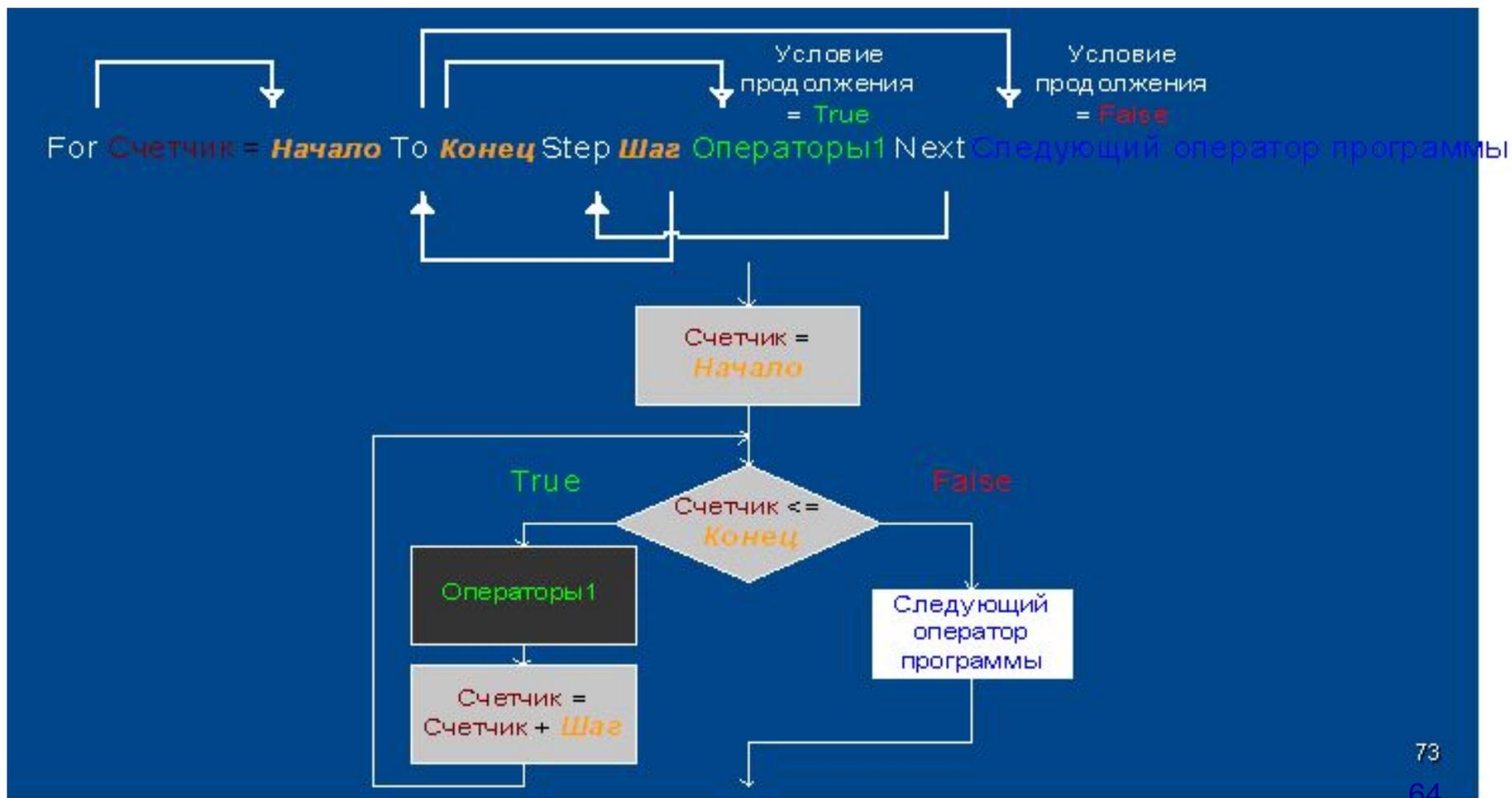
```
Case Else
```

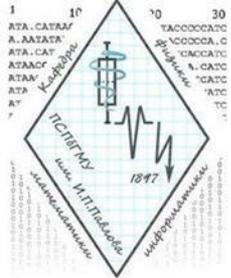
```
РежимРаботы = "Ошибка задания номера дня"
```

```
End Select
```



5.4. Семейство операторов For





Семейство операторов For

- Форматы оператора

For Счетчик=Начало To Конец [Step Шаг]

[Операторы]

[Exit For]

[Операторы]

Next [Счетчик]

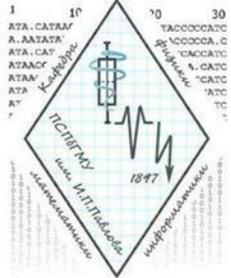
For Each Элемент In Группа

[Операторы]

[Exit For]

[Операторы]

Next [Элемент]



Семейство операторов For

Пример программы с оператором цикла for

```
Dim i As Integer, AA(10) As Double, _
```

```
BB(10) As Double, j As Variant
```

```
For i = 1 To 10
```

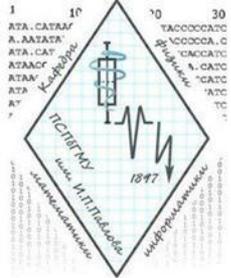
```
    AA(i) = i
```

```
Next i
```

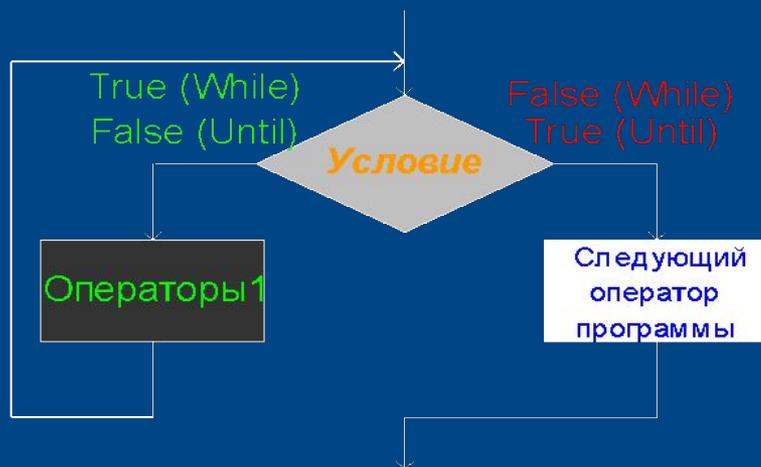
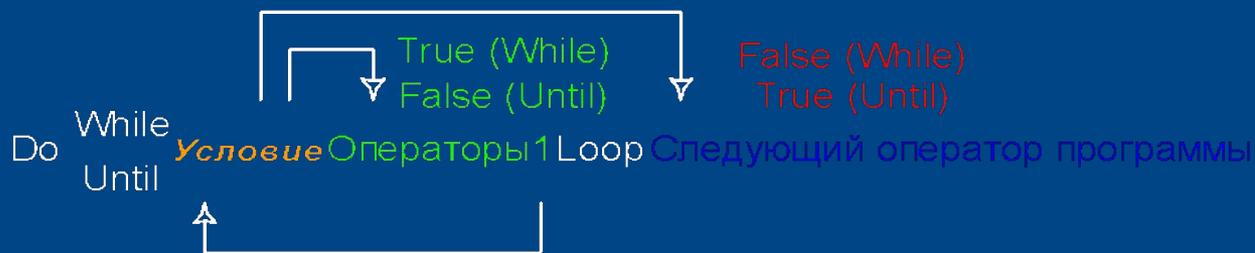
```
For Each j In AA
```

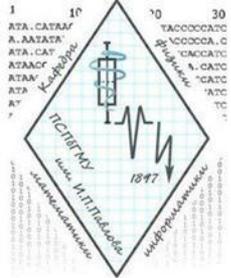
```
    BB(j) = AA(j)
```

```
Next j
```

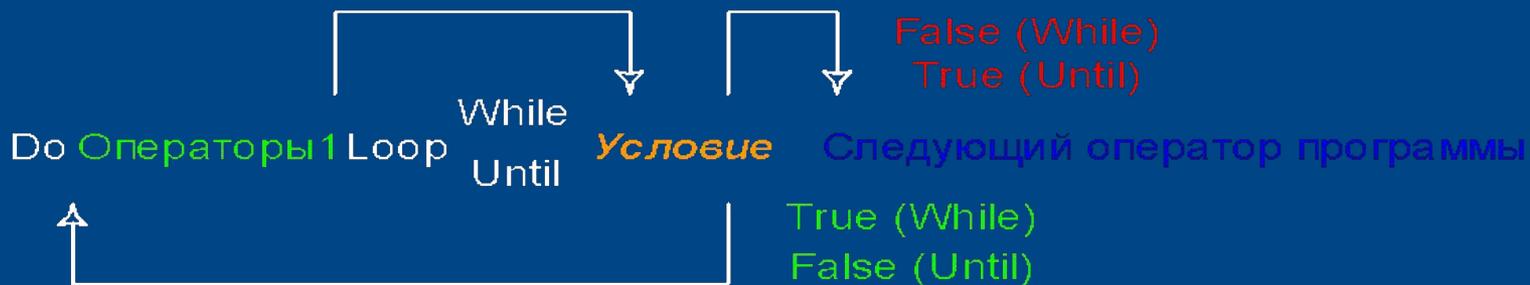


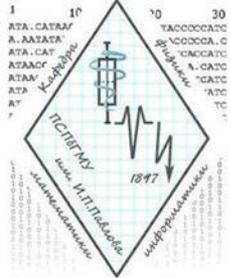
5.5. Семейство операторов Do





Семейство операторов Do





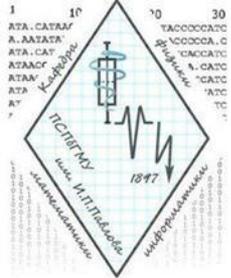
Семейство операторов Do

- Операторы While выполняются до тех пор, пока Условие = True.

или

Do [While Условие]
[Операторы]
[Exit Do]
[Операторы]
Loop

Do
[Операторы]
[Exit Do]
[Операторы]
Loop [While Условие]



Семейство операторов Do

Пример программы с оператором цикла While

```
Dim i As Integer, AA(10) As Double, _
```

```
BB(10) As Double, j As Variant
```

```
i = 1
```

```
Do While i <= 10
```

```
AA(i) = i
```

```
i = i + 1
```

```
Loop
```

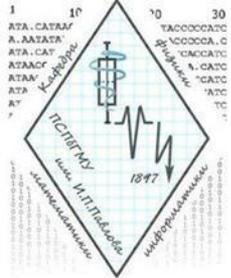
```
j = 1
```

```
Do
```

```
BB(j) = AA(j)
```

```
j = j + 1
```

```
Loop While j <= 10
```



Семейство операторов Do

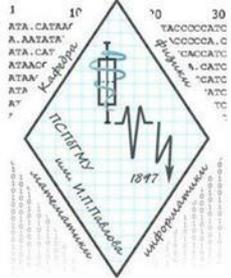
- Операторы Until выполняются до тех пор, пока Условие = False

Do [Until Условие]
[Операторы]
[Exit Do]
[Операторы]

ил
и

Do
[Операторы]
[Exit Do]
[Операторы]
Loop [Until Условие]

Loop



Семейство операторов Do

Пример программы с оператором цикла Until

```
Dim i As Integer, AA(10) As Double, _
```

```
BB(10) As Double, j As Variant
```

```
i = 1
```

```
Do Until i > 10
```

```
    AA(i) = i
```

```
    i = i + 1
```

```
Loop
```

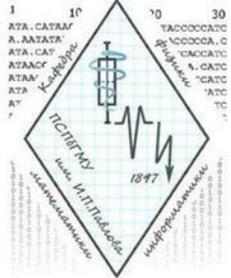
```
j = 1
```

```
Do
```

```
    BB(j) = AA(j)
```

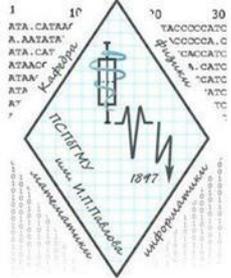
```
    j = j + 1
```

```
Loop Until j > 10
```



Семейство операторов Do

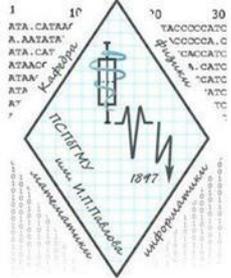
- Оператор выполняется, пока Условие = True
- While Условие
- [Операторы]
- Wend



6. Процедуры и функции

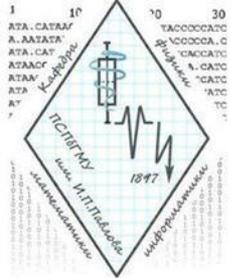
Рассматриваемые вопросы:

- Процедуры в VBA
- Функции в VBA
- Возврат значений из процедур и функций в вызывающую программу через список формальных параметров
- Встроенные функции
- Функции Excel, определенные пользователем



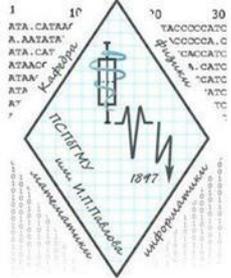
6.1. Процедуры в VBA

- Процедурой называется самостоятельная программа, предназначенная для решения определенной задачи.
- Каждая процедура имеет имя. Это имя является идентификатором процедуры.
- Макрос Excel представляет собой процедуру VBA.



Процедуры в VBA

- Каждая процедура может быть вызвана по имени. Если вызов отсутствует, то процедура выполняться не будет.
- Процедура может быть запущена на выполнения с помощью интегрированной среды отладки VBA.
- Каждая процедура имеет свои коды, которые должны быть оформлены заданным языком программирования способом.
- Для решения задачи процедура может потребовать набор аргументов (исходные данные), которые передаются ей в момент вызова.



Процедуры в VBA

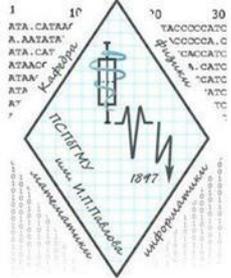
Формат описания списка аргументов:

[Optional] [ByVal или ByVal] [ParamArray]

ИмяПеременной[()] [As Тип]

[=поУмолчанию]

Optional - необязательный элемент. Должен иметь тип Variant. Все последующие элементы списка должны иметь такой же ключ и тип.



Процедуры в VBA

Формат описания списка аргументов:

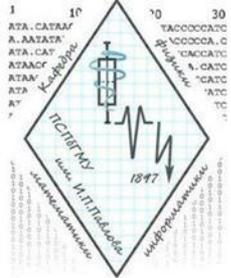
[Optional] [**ByVal** или **ByRef**] [ParamArray]

ИмяПеременной[()] [As Тип]

[=поУмолчанию]

Способ передачи параметров

ByVal – по значению, **ByRef** – по ссылке (по умолчанию)



Процедуры в VBA

Формат описания списка аргументов:

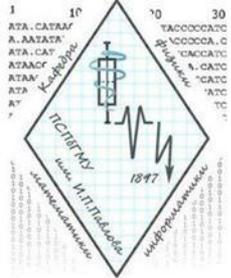
[Optional] [ByVal или ByVal] [**ParamArray**]

ИмяПеременной[()] [As Тип]

[=поУмолчанию]

ParamArray – неизвестное число параметров.

Может быть использовано только с последним элементом списка формальных параметров и позволяет передавать динамически объявляемый массив



Процедуры в VBA

Формат описания списка аргументов:

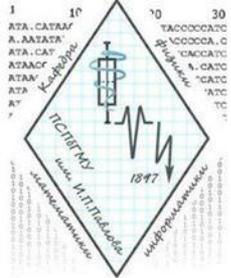
[Optional] [ByVal или ByVal] [ParamArray]

ИмяПеременной[()] [As Тип]

[=поУмолчанию]

ИмяПеременной[()] - аргумент процедуры (обычная переменная или массив).

Идентификатор, имеющий смысл формального параметра процедуры. Может быть несколько аргументов.



Процедуры в VBA

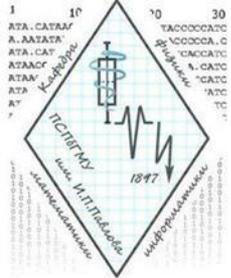
Формат описания списка аргументов:

[Optional] [ByVal или ByVal] [ParamArray]

ИмяПеременной[()] [As Тип]

[=поУмолчанию]

[As Тип] - тип данных аргумента



Процедуры в VBA

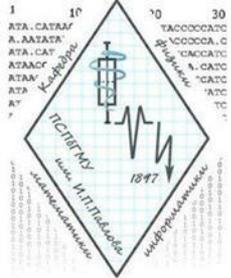
Формат описания списка аргументов:

[Optional] [ByVal или ByVal] [ParamArray]

ИмяПеременной[()] [As Тип]

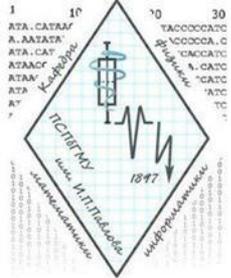
[=поУмолчанию]

[=поУмолчанию] - значение аргумента по умолчанию



Процедуры в VBA

- После заголовка процедуры следует конечное число обычных операторов языка VBA, представляющих собой тело определения функции. Если в их состав входят операторы объявления переменных Dim, то имеет место объявление собственных локальных переменных процедуры.
- Если в заголовке процедуры не указан ключ Static, то эти переменные не сохраняют свои значения между вызовами и каждый раз значения в них должны записываться заново.



Процедуры в VBA

Пример процедуры:

```
Sub ПримерПроцедуры(День As Integer, _ РежимРаботы  
As String)
```

```
    Select Case День
```

```
        Case 1
```

```
            РежимРаботы = "Прием документов"
```

```
        Case 2, 3, 4, 5
```

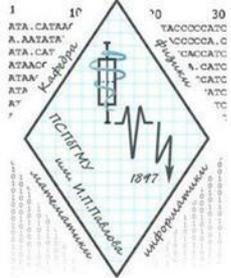
```
            РежимРаботы = "Выдача документов"
```

```
        Case Else
```

```
            РежимРаботы = "Выходные дни"
```

```
    End Select
```

```
End Sub
```



Процедуры в VBA

Пример вызывающей процедуры:

```
Sub ДемонстрацияПримераВызоваПроцедуры()
```

```
Dim a As Integer b As Integer , _
```

```
s As String, ss As String
```

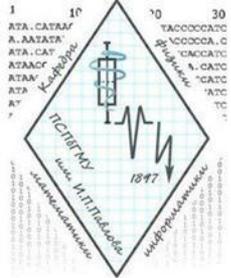
```
a = 1
```

```
Call ПримерПроцедуры(a, s)
```

```
b = 6
```

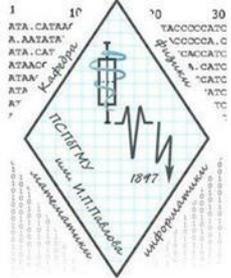
```
ПримерПроцедуры b, ss
```

```
End Sub
```



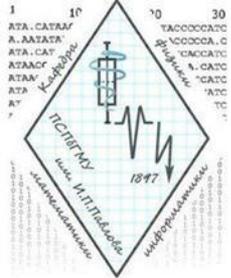
Процедуры в VBA

- Аргументами процедуры в момент ее описания являются так называемые *формальные параметры*. Они используются как полноправные участники любых операций и операторов тела процедуры для указания необходимой последовательности действий
- *Формальные параметры* получают физические адреса памяти для своего размещения они только в момент вызова процедуры. Обычно говорят, что процедура (функция) вызывается с *фактическими параметрами*
- Использование формальных параметров позволяет многократно вызывать процедуру из разных точек программы с различными аргументами.



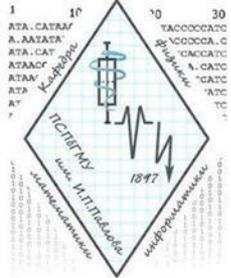
6.2. Функции в VBA

- Функцией называется самостоятельная вызываемая через оператор присваивания программа, предназначенная для решения определенной задачи.



Функции в VBA

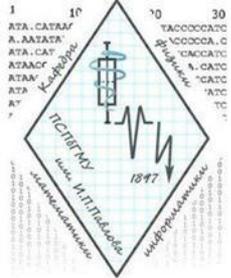
- [Public или Private] [Static] Function Имя [(СписокАргументов)] [As Тип]
- [Операторы]
- [Имя=Выражение]
- [Exit Function]
- [Операторы]
- [Имя=Выражение]
- End Function



Функции в VBA

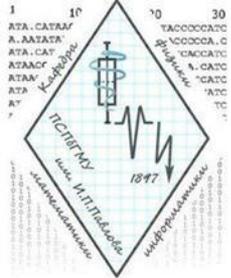
Формат описания списка аргументов:

[Optional] [ByVal или ByVal] [ParamArray]
ИмяПеременной[()] [As Тип] [=поУмолчанию]



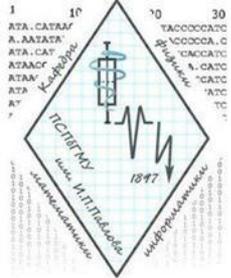
6.3. Возврат значений из процедур и функций в вызывающую программу через список формальных параметров

- При создании определений функций в языке VBA программисту предоставляют возможность передавать значения через список формальных параметров двумя способами: по значению `ByVal` и по ссылке `ByRef`.



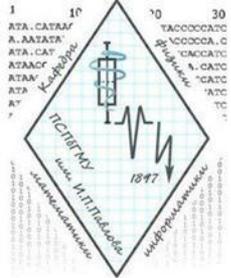
Возврат значений из процедур и функций в вызывающую программу через список формальных параметров

- Первый способ (по значению) является основным и выбирается исходя из соображений повышения надежности создаваемой программы. Если он используется, то вызываемая функция или процедура работает только с копией значения, находящегося в фактической ячейке вызывающей программы в момент вызова функции и не может изменить его. Поэтому в этом случае реализуется механизм односторонней передачи данных от вызывающей функции к вызываемой.



Возврат значений из процедур и функций в вызывающую программу через список формальных параметров

- Второй способ (по ссылке) отличается тем, что вызываемая функция получает в свое распоряжение не копию данных, а адрес ячейки памяти, в которой эти данные находятся. Как следствие, у вызываемой функции появляется возможность изменить содержимое ячеек вызывающей программы. Для этого в определении функции оператором присваивания задаются необходимые значения формальному параметру. В момент вызова процедуры (функции) формальному параметру ставится в соответствие фактическая ячейка памяти вызывающей программы. Именно в ней и произойдут указанные в определении изменения.



6.4. Встроенные функции

Cos Function

[See Also](#) [Example](#) [Specifics](#)

Returns a **Double** specifying the cosine of an angle.

Syntax

Cos(*number*)

The required *number* [argument](#) is a [Double](#) or any valid [numeric expression](#) that expresses an angle in radians.

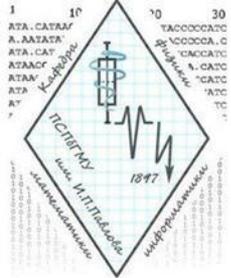
Remarks

The **Cos** function takes an angle and returns the ratio of two sides of a right triangle. The ratio is the length of the side adjacent to the angle divided by the length of the hypotenuse.

The result lies in the range -1 to 1.

To convert degrees to radians, multiply degrees by $\pi/180$. To convert radians to degrees, multiply radians by $180/\pi$.

Windows taskbar: Пуск, F:\Лекции, Документ..., 070306Пр..., Microsoft..., Microsoft..., Microsoft..., Microsoft..., 18:19



6.5. Функции Excel, определенные пользователем

'Функция, определенная пользователем

Function Расчет_налога(Начислено As _ Integer)

 Расчет_налога = Начислено * 0.12

End Function



vid.mp4

