

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**



**ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ**

**П.В. Бураков, В.Ю. Петров**

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ  
В ЭКОНОМИКЕ**

**Учебное пособие**



**Санкт-Петербург**

**2010**

Бураков Петр Васильевич, Петров Вадим Юрьевич. Информационные системы в экономике. Учебное пособие. – СПб: СПбГУ ИТМО, 2010. – 66с.

Учебное пособие позволяет студентам быстро овладеть навыками использования современных средств разработки приложений в среде Office для создания простых информационных систем. Рассмотрены начальные сведения о программировании на Visual Basic for Application (VBA), использовании встроенных функций.

Все примеры взяты из реальных разработок офисных программ для бухгалтеров, экономистов и инженеров.

Для студентов специальностей 080801 «Прикладная информатика в экономике»

Рекомендовано к печати на заседании ученого совета Гуманитарного факультета, протокол № 6 от 15/12/09



СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007-2008 годы и успешно реализовал инновационную образовательную программу «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий», что позволило выйти на качественно новый уровень подготовки выпускников и удовлетворять возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях науки. Реализация этой программы создала основу формирования программы дальнейшего развития вуза до 2015 года, включая внедрение современной модели образования.

©Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2010  
©П.В.Бураков, В.Ю. Петров., 2010

## ОГЛАВЛЕНИЕ

<b>Введение</b> .....	<b>5</b>
<b>1.Исследование встроенных функций Excel и запись макросов</b> .....	<b>6</b>
1. Основные положения.....	6
2. Исследование выполнения некоторых функций.....	7
2.1. Логические функции.....	7
2.2. Функции даты и времени.....	9
2.3. Математические функции.....	9
2.4. Функции просмотра и ссылок.....	11
3. Запись макросов.....	16
<b>2.Создание шаблонов документов (Word). Создание автоматизированного приложения (Excel)</b> .....	<b>18</b>
1. Создание шаблонов документов (Word).....	18
2. Создание автоматизированного приложения (Excel).....	21
2.1 Прямая задача.....	21
2.2 Обратная задача.....	23
<b>3.Технология создания бланков, используемых в экономических и бухгалтерских задачах</b> .....	<b>25</b>
1. Создание бланка платежного поручения.....	25
2. Задание для самостоятельной работы.....	30
<b>4.Исследование встроенных функций и проверка работы операций</b> .....	<b>31</b>
1. Основные положения.....	31
1.1. Основные окна, используемые при отладке.....	31
1.2. Управление печатью в окне «Проверка» с использованием разделителей.....	33
1.3. Управление печатью с помощью функции Format.....	34
2. Исследование работы строковых функций.....	36
3. Исследование работы функций из категории «Дата и время».....	39
4. Исследование работы логических операторов.....	41
5. Функции логических проверок.....	42
<b>5.Редактор VBA</b> .....	<b>44</b>
1. Интегрированная среда разработки. Общие положения.....	44
2. Окна редактора VBA.....	45
2.1. Окно проекта.....	45
2.2. Окно модуля.....	46
2.3. Окно форм и окно свойств.....	49
3. Создание простейших программ и экранных форм.....	49
4. Задание параметров редактора.....	51
<b>6.Создание функций, процедур и их отладка. Применение циклов и повторяемых структур</b> .....	<b>53</b>
1. Основные положения.....	53
2. Запуск и отладка процедур.....	54
3. Создание пользовательских функций.....	56

3.1 Определения и особенности создания пользовательских функций.....	56
3.2 Вызов пользовательских функций для использования в формулах рабочего листа.....	58
3.3 Аргументы пользовательских функций.....	59
3.4 Использование функций рабочего листа в процедурах и функциях VBA.....	61
4. Создание процедур.....	62
<b>Литература.....</b>	<b>64</b>

## ВВЕДЕНИЕ

Следует признать, что любая версия Microsoft Office, является совокупностью настольных приложений, ориентированных на огромный круг разнообразных пользователей, с успехом решающих свои многочисленные задачи. Во всех разработках большое внимание уделяется интеграции приложений, и каждый шаг разработчиков – это новое революционное позиционирование системы.

При всем этом, начиная с Office 95, офисная среда отражает тенденции развития информационных технологий и претерпевает изменения как платформа разработки программных решений. Встроенный язык программирования *Visual Basic for application (VBA)*, начиная с использования его в Excel 5, все более соответствует требованиям объектно-ориентированного программирования и универсальных языков высокого уровня и в совокупности с возможностями среды Office позволяет творить чудеса.

Программирование в среде Office называют «Офисным программированием». Успех создания приложений на современном этапе зависит, в конечном итоге, от того, насколько широко программисты и пользователи будут использовать платформу Office для построения своих решений. В связи с этим важным моментом является знание VBA и основ офисного программирования.

Пособие последовательно, шаг за шагом вводит пользователя в мир объектов Office, их методов, свойств, событий, знакомит с интегрированной средой разработки, технологией визуального программирования. В работе большое количество примеров и заданий, требуемых конкретного или альтернативного решения. Такой подход всегда полезней и информативней, чем разговоры о том, как создать программы.

Уделено место и обработке ошибок, что позволяет обойти программные сбои и сделать любую программу работоспособной. Особенно это важно в тех случаях, когда пользователь вводит неверные данные или ошибается в некоторых действиях.

Основное внимание в пособии уделено табличному (Word) и текстовому (Excel) процессорам, имеющим более зрелые модели объектов и наиболее востребованным с точки зрения офисного программирования.

# 1. ИССЛЕДОВАНИЕ ВСТРОЕННЫХ ФУНКЦИЙ EXCEL и ЗАПИСЬ МАКРОСОВ

## 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ.

Простые формулы в ячейки можно вводить вручную. Но при сложных вычислениях, а также для ускорения, упрощения организации вычислений, для сокращения числа ошибок следует использовать *Мастер функций*.

*Мастер функций* - это диалог, позволяющий выбрать и выполнить большинство из часто употребляемых, существующих функций. Все для выполнения этих функций в Excel уже готово и Вам необходимо только указать на их аргументы, а если точнее, адреса ячеек, где эти аргументы хранятся.

Во многих случаях в качестве аргументов можно использовать все те же встроенные функции пакета. Такой подход позволяет реализовать достаточно сложные алгоритмы и математические выкладки.

Для запуска мастера функций необходимо активизировать ту ячейку, в которую хотите поместить функцию (формулу), после чего:

- ✓ или выполнить команду *Формулы/Библиотека функций/Вставить функцию*
- ✓ или щелкнуть мышкой по кнопке **f<sub>x</sub>**, расположенной рядом со строкой формул

Третье условие: утверждение  $F2+F3=F4$ , т.е.  $4+6=5$  - это тоже "ЛОЖЬ"

Второе условие: утверждение  $D2+D3=D4$ , т.е.  $3+4=5$  - это бред ("ЛОЖЬ")

Первое условие, которое будет проверять ф-я "ИЛИ" - это равенство  $B2+B3=B4$   
Для данного набора чисел это равенство верное и результатом сравнения будет "ИСТИНА"

В ячейку H4 с помощью мастера ф-й помещена логическая функция "ИЛИ", проверяющая три условия. Текст функции имеет вид:  
 $=ИЛИ(B2+B3=B4;D2+D3=D4;F2+F3=F4)$   
Т.к. результатом проверки первого условия является "ИСТИНА", ф-я возвращает значение "ИСТИНА",

Рисунок 1.1 - Пример оформления выполненного задания.

**Что нужно сделать в работе?** Самостоятельно решите задачи и проверьте правильность работы функций, указанных ниже. Все функции выполните, следуя диалогам, которые предложит вам мастер функций. В том

случае, если что-то не получается, используйте справку в рабочей таблице мастера функций.

**Русский алфавит при наборе адресов ячеек не используйте !**

**Как оформлять работу ?** Все исследования оформите с помощью средств Excel, включая панель рисования. Все сделанное в этой, а также и других работах, должно быть записано на дискете. Новый материал д.б. законспектирован. Для примера, на рисунке 1.2 приведен вариант оформления работы для случая изучения логической функции «ИЛИ».

## 2. ИССЛЕДОВАНИЕ ВЫПОЛНЕНИЯ НЕКОТОРЫХ ФУНКЦИЙ

### 2.1. Логические функции

**Задание № 1.** С помощью электронных таблиц решите задачу в соответствии с приведенным алгоритмом.

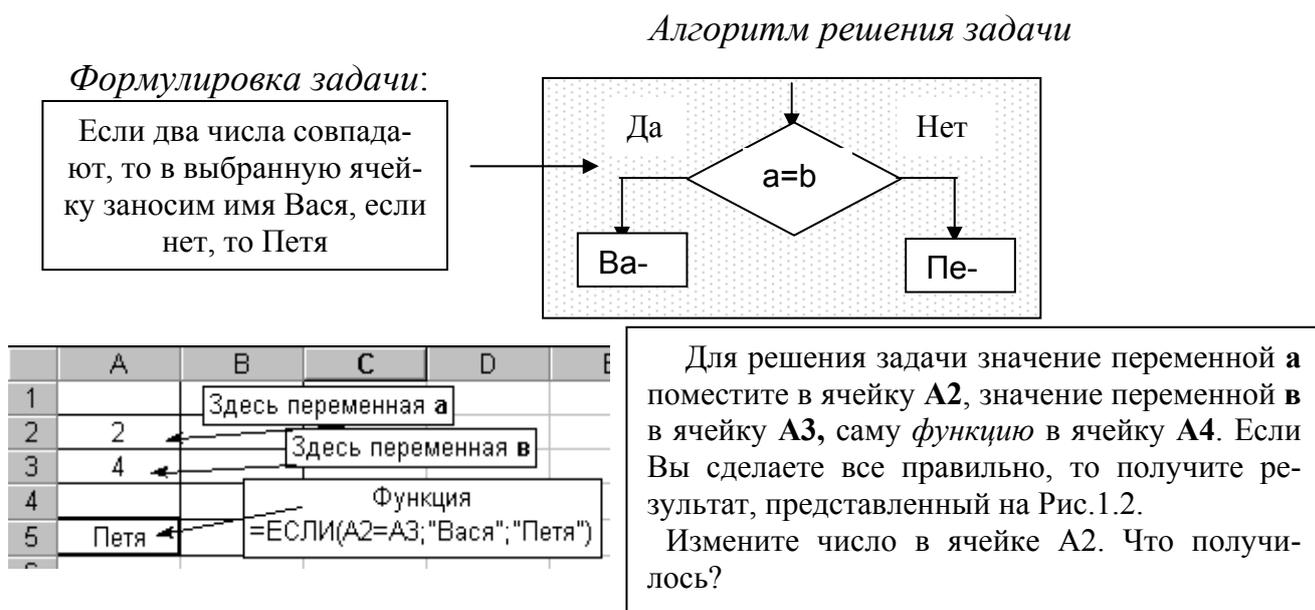


Рисунок 1.2 Пример оформления выполненного задания №1.

**Важная информация.** В задании 1 использовалась функция «ЕСЛИ». Отметим, что при наборе аргументов этой функции – (как впрочем, и других в последствии) адреса ячеек в поля ввода (рисунок 1.3) или в рабочие ячейки листа следует вводить щелкая по ним. Например (см. рис.1.3.), поле "Лог\_выражение" заполняем так: сначала щелкните по ячейке "А2", затем введите символ ">" с клавиатуры, щелчок по ячейке "А3".

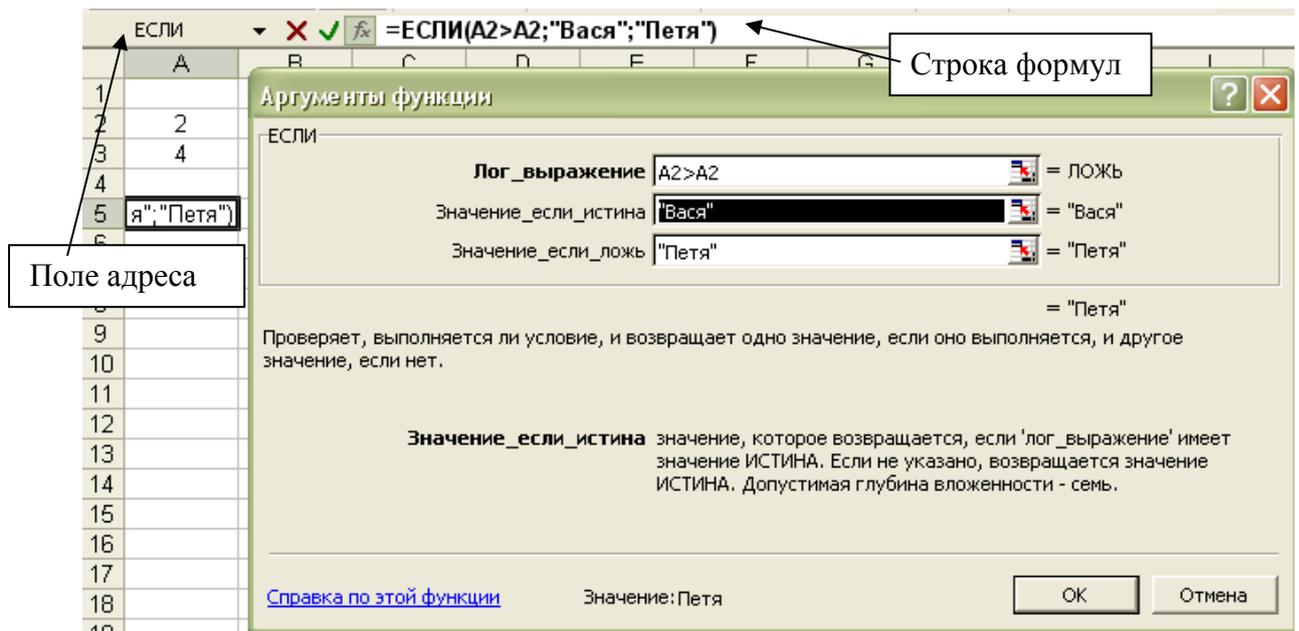


Рисунок 1.3. Ввод аргументов функции в рабочее окно мастера функций.

После заполнения всех полей окна «*Аргументы функции*» в *строке формул* появится текст формулы, а в *поле адреса*, слева от строки формул – название функции (см. рисунок 1.3.)

При этом в окне "*Аргументы функции*" есть еще масса полезной информации:

- ✓ сообщение о том, что делает функция;
- ✓ при щелчке по каждому полю для ввода аргументов выводится информация о том, что и в каком виде необходимо в него вводить;
- ✓ около каждого поля приводится возвращаемое им значение;
- ✓ выводится значение, возвращаемое функцией, причем, уже здесь при неверно введенной информации будет прописана ошибка;
- ✓ ! если щелкнуть по "*Справка по этой функции*" будет выведена не только справочная информация, но и примеры;
- ✓ ! если щелкнуть по полю для ввода аргумента, а после этого по *полю адреса*, то в качестве аргумента можно использовать любую из встроенных функций Excel. Следующий задание как раз и касается такой проблемы.

**Задание № 2.** В качестве аргумента функции «*ЕСЛИ*», можно использовать и ту же самую функцию «*ЕСЛИ*», и другие функции. Причем, уровень вложенности функций может быть достаточно велик. Это позволяет создавать функции для сложных алгоритмов.

Используя приведенный на рисунке 1.4 алгоритм, решите задачу с помощью электронных таблиц. Здесь Вам понадобится применить вложенные функции.

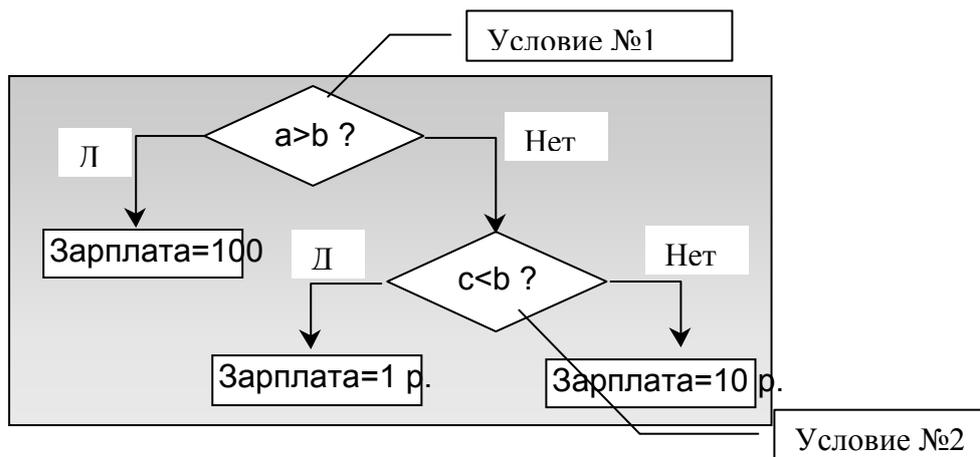


Рисунок 1.4 –Алгоритм для задания №2

## 2.2 Функции даты и времени.

	A	B	C	D	E	F	G	H
1								
2	03.08.99	←	3	←	=ДЕНЬНЕД(A2), оперделяет, каким по счету днем недели является дата , хранящаяся в			
3								
4	09.09.99	←	36	←	=ДНЕЙ360(A2;A4), определяет количество дней между двумя датами.			
5								
6	07.09.88	←	9	←	=МЕСЯЦ(A6), Оперделяет номер месяца в году , у даты из ячейки A6			
7								
8	В ячейки колонки A даты вводим сами. Это исходные данные		20.08.99		=СЕГОДНЯ(), вводит в ячейку сегодняшнюю дату.			
9								
10			6		=ДЕНЬНЕД(СЕГОДНЯ()), возвращает какой сегодня день недели. Здесь нужно использовать вложенные функции			
11								
12								
13								
14								

Рисунок 1.5 – Результаты выполнения задания №3

### Задание № 3

- ✓ Произведите вычисления, которые требуется выполнить на рисунке 1.5
- ✓ Определите, в какой день недели Вы родились?
- ✓ Определите, сколько дней Вы прожили от рождения до сегодняшнего дня?

## 2.3 Математические функции

### 2.3.1. Простые, часто встречающиеся функции.

Задание № 4 Самостоятельно исследуйте работу функций, таких как:

- ✓ **ЗНАК**, **ABS**, **ОСТАТ**, **ОТБР**, а также - *статистические*.
- ✓ **МИН**, **МАКС**
- ✓ исследуйте функцию **РАНГ**, которая определяет, какое именно место по возрастанию занимает ваше число среди массива чисел.

✓ Придумайте и приведите примеры использования функций округления чисел: **ОКРУГЛ, ЦЕЛОЕ, ОКРВВЕРХ, ОКРВНИЗ**. Чем они отличаются?

### 2.3.2. Функции СЧЕТЕСЛИ. СУММЕСЛИ

**Задание № 5.** Приведены данные, определяющие должности и оклады сотрудников предприятия, которые сведены в таблицу 1

Таблица 1 –Исходные данные

Фамилия	Должность	Оклад
Петров	инж	100
Иванов	тех	50
Кацман	менеджер	1000
Ли	бармен	20
Портная	инж	100
Худая	тех	70
Веселова	инж	120

Таблица 2 –Результирующие данные

Должность	Кол-во	Суммарный оклад
инж	3	320
тех	?	?
бармен	?	?

Ваша задача: так составить таблицу 2, чтобы в ней, исходя из данных таблицы 1, автоматически определялось:

- ✓ сколько по отдельности инженеров, техников и менеджеров работает на предприятии?
- ✓ сколько человек на предприятии имеет оклад менее 120?
- ✓ в примере используйте функции **СЧЕТЕСЛИ** и **СУММЕСЛИ**.

**Задание № 6.** Задана таблица с названиями фирм, к которым сегодня обращались клиенты, и текстом: «Успех», «Крах».

Таблица 3 - Вариант для задания №6

	А	В	С	Д	Е
1					
2		Дом +	Геоид	Нева	Техника
3		Геоид	Дом +	Нева	Геоид
4		Геоид	Геоид	Дом +	Нева
5		Нева	Техника	Нева	Дом +
6					
7			Успех	Крах	

Необходимо поместить в ячейку **Е7**, функцию, которая бы выполняла следующее:

- ✓ Считала бы количество встретившихся имен «Геоид» и «Нева».
- ✓ Сравнивала бы эти две суммы.

✓ Помещала бы в ячейку **Е7** слово, содержащееся в ячейке **С7**, если количество имен «Геоид» *больше* количества имен «Нева».

✓ Помещала бы в ячейку **Е7** слово, содержащееся в ячейке **Д7**, если количество имен «Геоид» *меньше* количества имен «Нева».

**В данном примере** Вам понадобится использовать несколько функций.

Можно, например, применить функции **ЕСЛИ** и **СЧЕТЕСЛИ**, но можно и какие-то другие. Сначала попробуйте выполнить отдельные задания, используя для выполнения каждого из них свою отдельную ячейку. В этом случае задача будет решена по частям, в несколько этапов.

После того, как будет получен положительный результат, синтезируйте требуемую функцию за один прием в одной ячейке. Результаты запомните.

## 2.4. Функции просмотра и ссылок

**2.4.1. Функция «ВЫБОР».** Возвращает одно значение из списка. На это значение указывает отдельный аргумент - номер индекса. (См. справку по функции)

**Задание № 7.** На чистом листе составьте таблицу с исходными данными, хранящимися в столбцах **В** и **С**, изображенную на рисунке 1.6. В столбце **D**, в указанные ячейки вставьте функции в соответствии с указаниями, содержащимися на том же рисунке.

	A	B	C	D	E	F	G	H
1	<b>Ведомость выдачи ЗП</b>							
2	<b>№ п.п</b>	<b>Список</b>	<b>Список_0</b>					
		<b>(Фамилия)</b>	<b>(Сумма)</b>					
3	1	Петров	100	2				
4	2	Иванов	200	Иванов				
5	3	Сидоров	300					
6	4	Уткин	400					
7	5	Гусев	500					
8	6	Собакин	600	100				
9	7	Свиньин	700	300				
10								
11	Сюда поместим функцию, которая суммирует премию с ЗП выбранного по индексу гражданина:							
12	=СУММ(D8,ВЫБОР(D3,C3,C4,C6,C7,C8,C9))							
13								
14								
15								

В эту ячейку будем вводить индекс. Для данного случая он совпал с № п.п. В принципе, индекс может быть и числом рассчитанным по формуле.

Функция, введенная в эту ячейку вернет фамилию гражданина в соответствии с введенным в ячейку D2 индексом.  
=ВЫБОР(D3,В3,В4,В5,В6,В7,В8,В9)

В этой ячейке находится сумма премии, которую мы добавим к зарплате гражданина, которого вычислим через индекс из ячейки D3.

Рисунок 1.6 – Исходные данные и результат выполнения задания №№ 7-8

**Задание № 8.** В списке на рисунке 1.6 содержится 7 фамилий.

Ваша задача: сделать так, чтобы из этого списка автоматический производился выбор дежурного на каждый день недели. Т.е. в понедельник - Петров, во вторник - Иванов и т.д. Здесь необходимо использовать функции **Даты и времени** и **ВЫБОР**.

### 2.4.2. Функция «ДВССЫЛ»

Используется для того, чтобы получить значение из ячейки по ссылке, которая находится в другой ячейке. Т. е., в данном случае, в процессе обмена задействованы три ячейки :

- **ячейка с функцией**, куда должна быть возвращена информация,
- **ячейка с информацией**, которая возвращается в ячейку с функцией
- **ячейка содержащая адрес ячейки с информацией**. (подробней см. справку).

Процесс обмена информацией для функции «Двойная ссылка» представлен на рисунке 1.7.

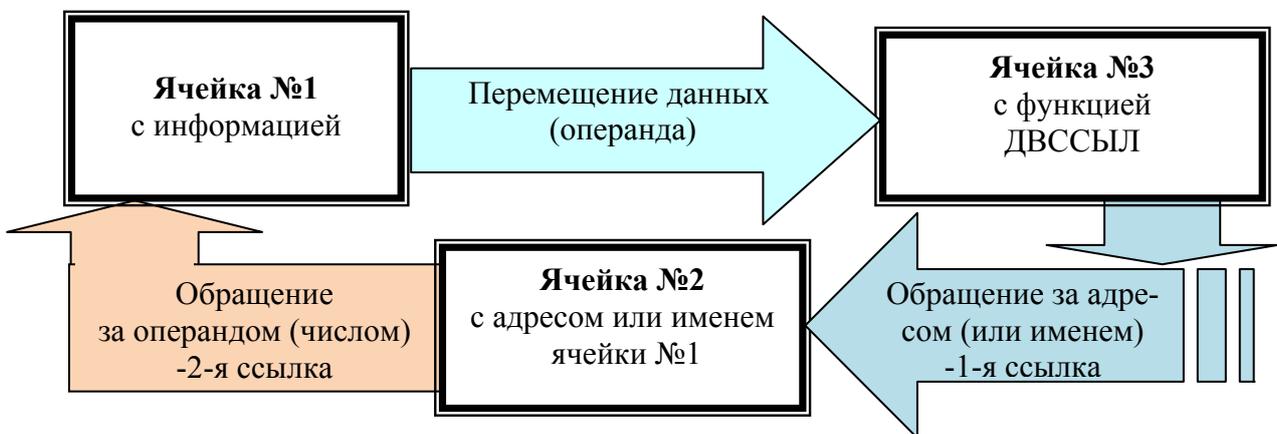


Рисунок 1.7- Перемещение данных при выполнении команды «ДВССЫЛ»

**Задание № 9.** У Вас есть огромная таблица (назовем ее основная), в которую собраны сведения о сотрудниках вашего предприятия, их фамилии, должности и оклады. Сотрудников много, должностей значительно меньше. Это значит, что сотрудников с одинаковыми должностями несколько. Причем каждой должности соответствует своя заработная плата.

Представьте, что каждой должности поставлена в соответствие новая заработная плата. Это повлечет за собой необходимость всем гражданам во всей таблице изменить величину заработной платы. Если предприятие большое это займет много времени.

**Ваша задача** автоматизировать процесс замены окладов у сотрудников.

#### **Конкретизируем эту задачу.**

Пусть «Основная таблица» для вашего предприятия представлена на рисунке 1.8. В нее включено всего шесть человек, которые могут занимать три должности. Создайте ее на чистом листе.

Для автоматизации процесса замены окладов у сотрудников, создадим вспомогательную таблицу с должностями и соответствующими им окладами, не касаясь фамилий. В принципе, эта таблица должна быть у вас в любом случае, поскольку несет в себе определяющие, во многих случаях, справочные данные. Пусть в ней будет всего 3 строчки - 3 должности и 3 оклада.

**Необходимо сделать так, чтобы:**

- ✓ смена оклада в этой вспомогательной таблице влекла за собой изменение окладов во всей большой таблице;
- ✓ изменение должности у сотрудника в основной таблице автоматически изменяло и его оклад.

В представленном варианте вы выигрываете всего в два раза. Но если предприятие большое, то выигрыш будет велик.

Для того чтобы добиться сказанного необходимо использовать функцию **ДВССЫЛ**, причем, со ссылкой по имени. Завершите изменения таблиц представленных на рисунке 1.8, и докажите, что они работоспособны.

	A	B	C	D	E	F	G
1	Основная таблица						
2	Гражданин	Должность	Оклад	В закрашенные ячейки вставьте функции ДВССЫЛ так, чтобы получить в них указанные результаты.  В закрашенные ячейки будем вводить любые оклады. При этом оклады в основной таблице для соответствующих должностей должны соответственно измениться. Ячейкам присвойте имена: "Матрос" "Солдат" "Санитарка"			
3	Денис	Матрос	100				
4	Вика	Солдат	200				
5	Оксана	Санитарка	300				
6	Алеша	Матрос	100				
7	Ваня	Солдат	200				
8	Катя	Солдат	200				
9							
10	Вспомогательная таблица						
11	№ п.п.	Должность	Оклад				
12	1	Матрос	100				
13	2	Солдат	200				
14	3	Санитарка	300				
15							

Рис. 1.8. Исходные данные и результат выполнения функции «ДВССЫЛ»

### 2.4.3 Функция «СМЕЩ»

Возвращает ссылку на диапазон, отстоящий от ячейки или диапазона ячеек на заданное число строк и столбцов. Возвращаемая ссылка может

быть отдельной ячейкой или диапазоном ячеек. Можно задавать количество возвращаемых строк и столбцов.

### Синтаксис

**СМЕЩ(ссылка; смещ\_по\_строкам; смещ\_по\_столбцам; высота;ширина)**

Эта функция хорошо прокомментирована в справке. Кроме того, как уже отмечалось, в справке есть примеры.

Здесь же отметим только то, что при описании функции (в справке либо в окне диалога) ее аргументы - высота и ширина представлены текстом с обычным начертанием, не полужирным. Это означает, что означенные параметры не являются обязательными и их можно опустить.

**Задание № 10** По справочным данным разберитесь с использованием функции «СМЕЩ» и приведите свои примеры ее использования.

При изучении работы этой функции удобнее всего заполнить поле из рядом стоящих ячеек (например, поле размером 5x5) разными цифрами. В центральную ячейку поля ввести функцию и проанализировать результат.

#### 2.4.4 Функция «ЯЧЕЙКА»

Это функция, которая позволяет получить массу полезных сведений о ячейке: адрес, номер строки или столбца, содержимое и т.д.. (см. справку).

**Задание № 11.** Приведите примеры использования функции «ЯЧЕЙКА».

#### 2.4.5. Функция «ИНДЕКС»

Возвращает значение элемента таблицы или массива.

Функция ИНДЕКС имеет две синтаксические формы: ссылка и массив.

Для примера будем использовать форму массива, которая возвращает значение или массив значений. Ее синтаксис имеет вид:.

**ИНДЕКС(массив;номер\_строки;номер\_столбца)**

#### **Задание № 12.**

✓ Изучите самостоятельно справочные данные для функции «ИНДЕКС».

✓ Создайте простейшую справочную систему, используя функцию «ИНДЕКС», в основу которой положены две таблицы, представленные на рисунке 1.9.

Пусть единая тарифная сетка задана в виде таблицы. Она должна быть скрытой от пользователей. Для общения с пользователями, т.е. для того, чтобы пользователь имел возможность ввести свой разряд и получить от машины тарифный коэффициент необходимо создать справочное табло.

Это табло представляет собой электронную таблицу, с двумя рабочими ячейками: в одну из которых граждане вводят свой разряд, а в другой получают тарифный коэффициент, который машина должна найти в скрытой таблице.

Скрытость таблицы не следует понимать формально в данной задаче.

Справочное табло	
Введи разряд	2
Данному разряду соответствует тарифный коэффициент	1,5

Скрытая таблица	
Разряд	Коэффициент
1	1
2	1,5
3	2
4	3
5	4
6	4,5

Рисунок 1.9 - Исследование работы функции «ИНДЕКС»

#### 2.4.6 Функция «ВПР»

Ищет значение в крайнем левом столбце таблицы и возвращает значение в найденной строке из указанного столбца таблицы.

По умолчанию таблица должна быть отсортирована по возрастанию.

	A	B	C	D	E	F	G	H
1								
2		Таблица				=ВПР(3;B3:E8;4) - это пример использования ВПР		
3		1	3	5	30			
4		2	6	33	20		100	
5		3	2	55	100			
6		5	9	44	40			
7		10	8	11	50		9	
8		13	100	2	70		50	
9								
10				Ячейка с функцией ВПР		Ячейка для ввода		
11	В ячейку G8 поместите функцию "ВПР". Она должна выбрать необходимые данные из							
12	закрашенной части таблицы. Причем так, чтобы искомое значение, которое будем							
13	искать в первом столбце закрашенной таблицы, можно было бы вводить в ячейку G7							
14								

Рисунок 1.10 – Исследование работы функции «ВПР»

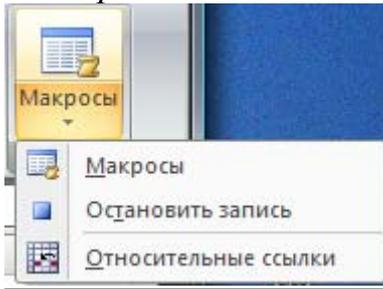
**Задание № 13**

- ✓ Выполните действия указанные на рисунке 1.10
- ✓ Выполните действия указанные на рисунке 1.10., используя функцию «ГПР»

**3. ЗАПИСЬ МАКРОСОВ**

Макрос это программный код, написанный на языке Visual Basic for Application (VBA). В ранних версиях Excel - 5 и 7 версиях- он записывался на отдельных листах. Для версий Excel 97 и выше макрос создается в редакторе программ VBA. По классификации макрос это такой программный код, который создал Макрорекодер.

Макрорекодер записывает автоматически все действия, произведенные в Excel в виде программного кода и запоминает его как отдельную процедуру - Макрос. Для запуска Макрорекодер следует выполнить команду *Вид/Макросы/Запись макроса* ( для старых версий - *Сервис/Макрос/Начать запись*).



Для остановки процесса записи макроса макрорекодером выполните команду «*Остановить запись*» с вкладки макросы. Нижняя кнопка на этой панели включает режим относительных или абсолютных ссылок.

**Задание № 14.**

Запустите табличный процессор Excel.

Включите макрорекодер.

После присвоения макросу имени, на рабочем листе активизируйте ячейку A3. В этой ячейке напишите любую фразу, например «Дай миллион».

Перейдите к ячейке B2.

Остановите запись.

Перейдите к *Модулю*, в котором находится записанный текст программы. Убедитесь, что программа существует. Для этого следует:

- ✓ перейти в редактор VBA - щелкните на клавиатуре по клавишам <Alt +F11>,
- ✓ в редакторе вызвать вызовите окно проекта – клавиши <Ctrl+R>,
- ✓ в окне проекта найти строку Modules/Module1 и щелкнуть по ней мышью,
- ✓ справа в рабочем окне модуля просмотреть программный код.

Запустите макрос на другом листе Excel, используя команду *Вид/Макросы/Макросы/имя макроса в рабочем окне*, и убедитесь, что он выполнен.

**Задание № 15.** Создайте еще один макрос, но с относительными ссылками. Ответьте, чем он отличается от первого, записанного (по умолчанию) с абсолютными ссылками.

**Задание № 16** Действуя аналогично, создайте макрос в редакторе Word. Присвойте макрос сочетанию клавиш и проверьте его работоспособность.

**Задание № 17** Приведите пример использования относительных и абсолютных адресов. Покажите когда целесообразно использовать абсолютную адресацию, а когда относительную.

## 2.СОЗДАНИЕ ШАБЛОНОВ ДОКУМЕНТОВ (WORD) СОЗДАНИЕ АВТОМАТИЗИРОВАННОГО ПРИЛОЖЕНИЯ (EXCEL)

### 1. СОЗДАНИЕ ШАБЛОНОВ ДОКУМЕНТОВ (WORD)

**Пояснительная записка** - приложение к протоколу согласования цены на создание научно-технической продукции, для работы, выполняемой в 2 этапа, представлена на рисунке 2.1.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
(расчет цены)

К протоколу № **ПОЛЕ НОМЕРА** от " \_\_\_ " **ПОЛЕ МЕСЯЦА** 199\_\_ г.  
согласования цены на научно-техническую продукцию специального назначения

**ПОЛЕ ВИДА РАБОТ.** **ПОЛЕ ГОДА**  
(указать вид работы: НИР, ОКР и т.п.)

по теме " **ПОЛЕ ИМЕНИ ТЕМЫ** "

---

Расчет стоимости работы произведен в соответствии с "Методикой определения цены НТП " на 2004 г., согласованной с **ПОЛЕ ИМЕНИ ПРЕДПРИЯТИЯ**

Плановая трудоемкость темы ( $T_{П}$ ) составляет:

для в/с ( $T_{ПВ}$ ): 980,2 ч/час, в т.ч. 1 этап -849,5 ч/час, 2-й этап -130,7 ч/час.  
для служащих ( $T_{ПШ}$ ): 980,2 ч/час, в т.ч. 1 этап -849,5 ч/час, 2-й этап -130,7 ч/час.

Денежное довольствие военнослужащих (в/с), непосредственно работающих по тематике составляет

$$ДДФ=(ДД + 0.776*ДД)* T_{ПВ}$$

где: ДД=11.36 - средняя стоимость часа в/с (руб);  
0.776 - коэффициент ФМП, относимый на себестоимость.

Денежное довольствие военнослужащих за непраработанное (неявочное время)

$$ДДД=ДД*0.455* T_{ПВ}$$

Расходы на оплату труда (РОТ) составляют

$$РОТ=(ЗП+ЗП*0.686)* T_{ПШ}$$

где: ЗП=4,12 - средняя стоимость часа в/с (руб);  
0,686- коэффициент ФМП, относимый на себестоимость.

Расходы на дополнительную оплату труда составляют

$$ДОТ= ЗП*0.153* T_{ПШ}$$

Отчисления на соц. определяются как:

$$ОСС=(РОТ+ДОТ)*0,385$$

Накладные расходы  $N_p=(ДДФ+РОТ)*1.592$ .

Прибыль  $P_p=9.1\%$  от себестоимости НТП.

НО ПНИР =**ПОЛЕ ИМЕНИ-1**=  
Руководитель работы =**ПОЛЕ ИМЕНИ-2**=

Рисунок 2.1.- Вид листа пояснительной записки.

Представленный документ содержит основной текст, который не изменяется при составлении следующих копий документа, а также текст, который может изменяться и для каждого документа будет свой. Именно этот текст и несет исходные данные и оригинальную информацию. Это: номер протокола, имя темы, даты и т.д.

В тех местах образца пояснительной записки, где должна располагаться изменяющаяся информация, текст написан полужирным курсивом. Например, в ***ПОЛЕ НОМЕРА*** будет располагаться номер протокола и т.д.

Задача по созданию шаблона сводится к формированию такого документа, в котором можно изменить лишь ту информацию, которая находится в указанных полях.

### Задание № 1

1. ***Создайте пояснительную записку*** в соответствии с образцом (см. рисунок 2.1).

Она занимает 1 лист и, в целом, набирается шрифтом Times New Roman Cyr, величиной 12пт. Линии проводятся с использованием панели рисования. Набирая основной текст, оставляйте места для отмеченных *полей ввода*.

### 2. ***Создайте поля ввода.***

Такие поля являются элементами управления с вкладки «Разработчик», а именно: «Текст» и «Поле со списком».

Для создания элемента управления **Текст** (в Word 2007) (в более ранних версиях Word этот элемент управления называется **поле формы**) сначала выполните команду *Настройка панели быстрого доступа*. Это можно сделать предварительно щелкнув правой кнопкой мыши по панели быстрого доступа. Далее, выбрав в рабочем окне «параметры Word», категорию «Основные» поднимите флажок «Показывать вкладку разработчик на ленте».

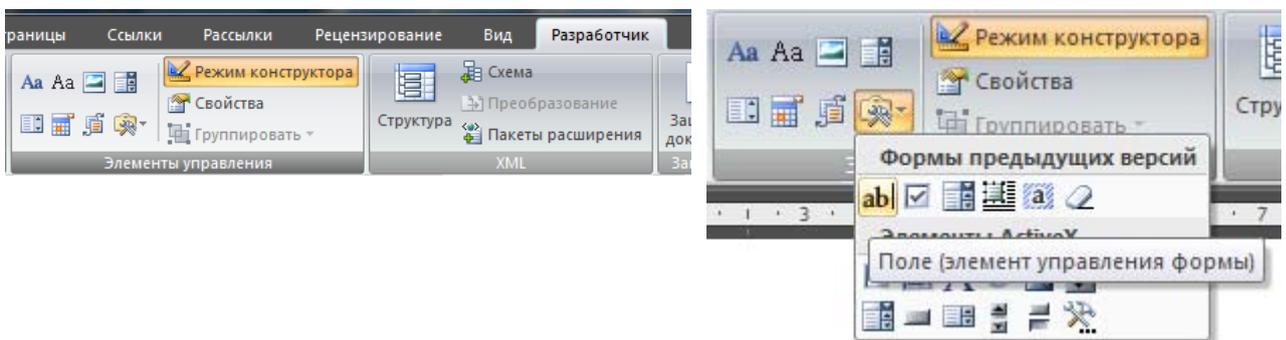


Рисунок 2.2. - Вид вкладки разработчик с элементами управления.

На ленте появится соответствующая вкладка с элементами управления, показанная на рисунке 2.2.. Если необходимо воспользоваться элементами предыдущих версий Office, выберите значок «*Формы предыдущих версий*».

Перетащите мышью выбранный элемент управления на рабочий лист.

**ПОЛЕ НОМЕРА** **ВВОДИМЫЙ ТЕКСТ** **ПОЛЕ НОМЕРА** -для Word 2007

 -для Word 2003

Теги для Word 2007 будут убраны при печати, просмотре печати, выключении режима конструктор. В этом же режиме возможно удаление полей.

Примечание: для ранних версий Word тоже самое можно сделать, используя команду меню *Вид/Панели инструментов/Формы*.

Свойства (параметры) созданного элемента управления можно установить после его активизации, выполнив команду «Свойства» со вкладки «Элементы управления». На рисунке 2.3 ниже продемонстрировано рабочее окно для установки свойств, а также вид строки с элементом управления «Текст» для трех случаев: вставка элемента управления; активизация элемента управления у которого введено название; окончательный вид поля после ввода в него текста «Научная работа».

По·теме·Место·для·ввода·текста.¶

По·теме·Место·для·ввода·текста.¶

¶

По·теме·Научная·работа¶

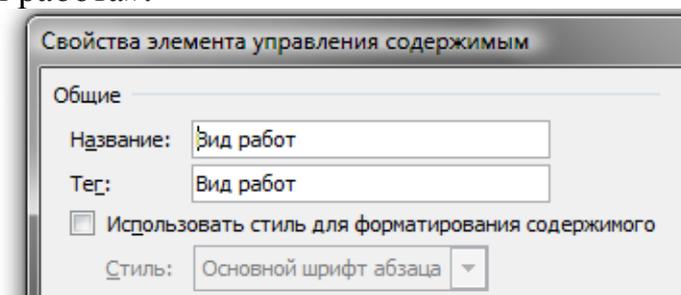


Рисунок 2.3. – Вид элемента управления «Текст»

При задании свойств форм используйте необходимые данные таблиц № 2.1 и № 2.2, приведенных ниже.

Таблица № 2.1

<i>Имена текстовых полей</i>	<i>Тип</i>	<i>Макс. Длина</i>	<i>Текст по умолчанию</i>
Поле номера	Число	3	123
Поле даты-1	Число	2	12
Поле года	Число	1	8
Поле вида работ	Обычный текст	80	НИР
Поле имени темы	Обычный текст	60	«МОРСКОЙ БОЙ»
Поле имени-1	Обычный текст	20	«Фамилия И.О.»
Поле имени-2	Обычный текст	60	«Иванов И.И.»

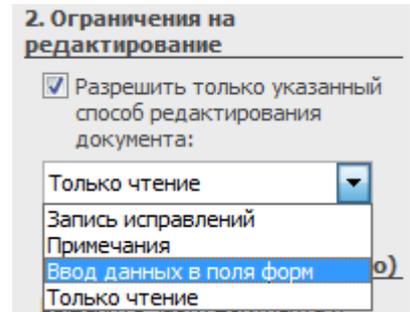
Таблица № 2.2

<i>Имена полей со списком</i>	<i>Элементы списка</i>
Поле месяца	Январь, Февраль, Март, Апрель, Май, Июнь, Июль, Август, Сентябрь, Октябрь, Ноябрь, Декабрь
Поле имени предприятия	Кировский завод, Малахит, ЛИТМО

### 3. Установите защиту документа.

В окне «Свойства» поставьте блокировку на удаление элемента.

Для этого выполните команду «Защитить документ» из вкладки «Разработчик». Включите режим «Ограничить форматирование и редактирование», в появившемся рабочем окне выберите группу команд «Ограничения на редактирование» – «Ввод данных в поля форм», после чего, не покидая это рабочее окно, включите защиту



Примечание: для ранних версий Word защита устанавливается командой меню *Сервис/Установить защиту.../С запретом любых изменений кроме как в поля форм*. После этой команды ввод данных станет возможным только в поля форм.

### 4. Запомните файл с именем «Pois\_Zap».

5. *Попробуйте ввести* в созданный документ информацию. Переход между полями производится кнопкой табуляции на клавиатуре или мышью. Сохраните его.

## 2. СОЗДАНИЕ АВТОМАТИЗИРОВАННОГО ПРИЛОЖЕНИЯ (EXCEL)

В соответствии с поступившей пояснительной запиской по расчету цены необходимо произвести такой расчет. Очевидно, что Excel идеально подходит для этих целей. Табличный процессор не только сформирует готовый бланк расчета, но и позволит при желании автоматически пересчитать результат при изменении какого-нибудь из них.

При этом возможны две задачи: прямая и обратная.

### 2.1. Прямая задача.

Эта задача предполагает, что все расчеты производятся в соответствии с данными пояснительной записки. В результате расчета получим стоимость работ по этапам и общую стоимость.

Примерный бланк для расчета структуры цены представлен на рисунке 2.4.

Денежные суммы по п.п. 6 (расходы на сл.командировки), 7 (прямые прочие расходы) и 11 (затраты по работам.....), которые выделены полужирным курсивом, задают исходя из нужд предприятия. Их задайте произвольно

	A	B	C	D	E
1					
2					
3	<b>СТРУКТУРА ЦЕНЫ</b>				
4	<b>на научно-техническое сопровождение</b>				
5	<b>по теме: "МОРСКОЙ БОЙ"</b>				
6					
7	<b>№ п.п</b>	<b>Наименование статей расходов</b>	<b>Всего</b>	<b>В т.ч. по этапам</b>	
8			<b>(руб)</b>	<b>1-й этап</b>	<b>2-й этап</b>
9	1	Денежное довольствие военнослужащих непосредственно работающ по тематике	13 298	7 979	5 319
10	2	Денежное довольствие военнослужащих за непроработанное (неявочное) время (ДДФ)	3 407	2 044	1 363
11	3	Расходы на оплату труда (РОТ)	12 207	7 324	4 883
12	4	Дополнительная заработная плата	1 108	665	443
13	5	Отчисления на соц. страхование	5 126	3 076	2 051
14	<b>6</b>	<b>Расходы на служебные командировки</b>	0	0	0
15	<b>7</b>	<b>Прямые прочие расходы</b>	0	0	0
16	8	Накладные расходы	40 604	24 362	16 242
17	9	ИТОГО собственные расходы	75 750	45 450	30 300
18	10	Прибыль	7 583	4 550	3 033
19	<b>11</b>	<b>Затраты по работам выполняемым сторонними организациями</b>	0	0	0
20	12	Цена	83 333	50 000	33 333
21	13	НДС	16 667	10 000	6 667
22	14	Стоимость работы	100 000	60 000	40 000
23					
24		<b>Начальник ОПНИР</b>			<b>С.Иванов</b>
25					
26		<b>Руководитель работы</b>			<b>В.Сидоров</b>

Рисунок 2.4 – Бланк расчета структуры цены.

При расчетах следует принять (см. рис.2.2 и номера статей):

*Итого собственные расходы (9) = 1+2+3+4+5+6+7+8;*

*Денежное довольствие в собственных расходах = 1+2+3+4+5+8*

**Задание №2** Создайте бланк для расчета структуры цены.. Для расчетов и формирования бланка следует выделить отдельный лист для ввода исходных данных: коэффициентов, дат, отдельных сумм и т.д. Непосредственно в ячейках бланка расположите необходимые формулы.

## 2.2. Обратная задача.

Обратная задача сводится к расчету структуры цены, исходя из того, что заданными являются данные для статей, которые выделены на рисунке 2.4 полужирным курсивом и п.п.14 (стоимость работы), а также процент денежного довольствия военнослужащих в собственных расходах по теме. Таким образом, исходя из стоимости работы, необходимо определить суммы статей расходов.

В расчетах используйте также данные пояснительной записки. При этом, для каждого из этапов, по заданным величинам, составив систему уравнений, определите сначала плановую трудоемкость, а потом уж и все статьи расходов.

На листе для ввода исходных данных и определения значений промежуточных величин предусмотрите поля для ввода переменных, примерно так, как это приведено на рисунке 2.5.

	А	В	С	Д	Е	Ф	Г
1	Руководитель работ		<i>В.Сидоров</i>	по теме	<i>"МОРСКОЙ БОЙ"</i>		
2	Начальник ОПНИР		<i>С.Иванов</i>				
3							
4	Параметры		Всего	1 этап	2 этап		
5	Стоимость работ (вводим сами)		100000	<b>60000</b>	<b>40000</b>		
6	Расходы на служ. командиров.		0	<b>0</b>	<b>0</b>		
7	Прочие прямые расходы		0	<b>0</b>	<b>0</b>		
8	Затраты на сторонние организации		0	<b>0</b>	<b>0</b>		
9	Процент денежного дов.		%	<b>50</b>	<b>50</b>		
10	Плановая тр-сть только военные (считает машина)	Т <sub>пв</sub> =	1318,238	395,470	263,649		
11	Плановая тр-сть только гражданские (считает машина)	Т <sub>пг</sub> =	131,962	382,411	254,943		
12							
13							
14	K <sub>dd</sub> =	<b>11,36</b>	Ср. стоимость часа в/с (руб)				
15	K <sub>1,592</sub> =	<b>1,592</b>	Козф. для расч. накл расходов				
16	K <sub>0,455</sub> =	<b>0,455</b>	Козф. расч. денеж. дов. за неявочное вр.				
17	K <sub>0,776</sub> =	<b>0,776</b>	Козф. ФМП, относимый на себестоимость				
18	K <sub>0,091</sub> =	<b>0,091</b>	Процент для рприбыли				
19							
20	K <sub>зр</sub> =	<b>11,36</b>	Ср. стоимость часа гражданских (руб)				
21	K <sub>0,686</sub> =	<b>0,686</b>	Козф. ФМП, относимый на себестоимость				
22	K <sub>0,153</sub> =	<b>0,153</b>	Козф. расч. доп зар. платы				
23	K <sub>0,385</sub> =	<b>0,385</b>	Соц. страх				

Данные в ячейках для непосредственного ввода набраны жирным курсивом

Рисунок 2.5 – Лист с исходными данными.

### 3. ТЕХНОЛОГИЯ СОЗДАНИЯ БЛАНКОВ, ИСПОЛЬЗУЕМЫХ В ЭКОНОМИЧЕСКИХ И БУХГАЛТЕРСКИХ ЗАДАЧАХ

В этой части пособия преследуются следующие цели:

1. Научиться составлять рабочие таблицы.
2. Научиться формировать бланки, практически любой степени сложности.
3. Размещать дополнительные кнопки на панелях инструментов.
4. Блокировать (запрещать запись в) отдельные ячейки на листе.

#### 1. СОЗДАНИЕ БЛАНКА ПЛАТЕЖНОГО ПОРУЧЕНИЯ

Типовой бланк *Платежного поручения (платежки)*, который должен получиться имеет вид представленный на рисунке 3.1.

	A	B	C	D	E	F	G	H	I	J	K	L	N	O	P	Q	R	S	T	U	V	W	X	Y
1																								
2		12/12/98		12 октября 2004 г.																				
3		Поступ. в банк плат.		Списано со сч. плат.																		0401060		
4																								
5		ПЛАТЕЖНОЕ ПОРУЧЕНИЕ № 11111		08 сентября 1998 г.		Почтой																		
6								Дата																
7																								1
8																								
9		Сумма	Пятьсот рублей 00 копеек		Пятьсот рублей 00 копеек		Пятьсот рублей 00 копеек																	
10		прописью																						
11																								
12		ИНН	1234567890	КПП	1234567	Сумма	500=00																	
14		ЗАО "ВАСЯ и ПЕТЯ и Компания "																						
15																								
16																								
17																								
18		Платательщик								Сч. №	1234567890123456789012													
20		Открытое Акционерное общество								БИК	1321													
21		"Промышленно строительный банк "																						
22																								
23		Банк плательщика								Сч. №	12345678901234111													
25		Закрытое Акционерное общество								БИК	123													
26		"Промышленно строительный банк "																						
27																								
28		Банк получателя								Сч. №	1234567890123456789012													
30		ИНН	1234567890	КПП						Сч. №	1234567890123456789012													
32		АОЗТ "Гуси илебеди , проживающие в районе "																						
33		Фирма г .Москвы																						
34										Вид оп	31	Срок. плат.	12/02/98											
35										Наз. пл		Очер. плат.	33											
36		Получатель								Код		Рез. поле												
38		12		12		12	12					21		12										
40		Возврат ссуды по договору произвести тов. Остатку на дом. по договору																						
41																								
42																								
43																								
44																								
45		Назначение платежа																						
47																								
48																								
49																								
50																								
51																								
52																								
53																								
54																								

Рисунок 3.1- бланк платежного поручения.

**Для создания бланка Платежного поручения выполните следующие действия:**

1. Установите параметры страницы:
  - ✓ ориентация – книжная;
  - ✓ поля: верхнее -1,5; нижнее -2,5; правое – 0,5; левое – 2;
  - ✓ колонтитулы должны отсутствовать (быть пустыми).
  
2. Установите высоту строк и ширину столбцов, участвующих в формировании «Платёжки» в соответствии с данными таблиц № 3.1 и № 3.2

Таблица №3.1

<b>Ширина столбцов</b>							
Имя столбца	Ширина столбца	Имя столбца	Ширина столбца	Имя столбца	Ширина столбца	Имя столбца	Ширина столбца
A	0,42	G	1,57	M	0,17	S	0,92
B	9,86	H	4,43	N	1,71	T	2,43
C	11,71	I	6,71	O	3,29	U	2,57
D	2,57	J	0,42	P	3,43	V	0,83
E	4,86	K	4,57	Q	4,57	W	3
F	4	L	1,57	R	4,57	X	0,33

Таблица № 3.2

<b>Высота строк</b>							
Номер строки	Высота строки	Номер строки	Высота строки	Номер строки	Высота строки	Номер строки	Высота строки
1	12,75	14	12,75	27	7	40	12,75
2	12,75	15	12,75	28	14	41	12,75
3	12,75	16	12,75	29	3	42	12,75
4	8,25	17	12,75	30	12,75	43	12,75
5	9,75	18	12,75	31	5,25	44	12,75
6	9	19	5	32	12,75	45	12,75
7	10,5	20	14	33	12,75	46-48	12,75
8	9,75	21	7	34	14	49	9,75
9	14,25	22	7	35	14	50	12,75
10	13	23	14	36	12,75	51	11,25
11	12,75	24	3	37	2,25	52	11,25
12	5	25	14	38	14,25	53	16,5
13	12,75	26	7	39	5	54	12,75

3. Выделите *весь лист* и установите шрифт «*Times New Roman Cyr*», *Полужирный*, *размером 10*, а формат для всех ячеек - «*Текстовый*».

4. Используя кнопку-«Объединить и поместить в центре», ( или любым другим способом ) отформатируйте указанные ниже ячейки:

B2:C2, B3:C3, E2:I2, E3:I3, I6:N6,  
P6:R6,U3:W4 K12:L12, K17:L17, K20:L21, K22:L23, K25:L26, K27:L28,  
K30:L30, B21:I22, B26:I27, N20:U21, N22:U23 N25:U26, N27:U28,  
W6:W7



Последние пять объединений, выделенные курсивом, отформатируйте, щелкнув дополнительно по кнопке «Выровнять по левому краю».



5. В соответствии с Таблицей № 3.3, в указанные ячейки введите текст, используя начертание для шрифта – обычный (не жирный):

Таблица № 3.3

<i>Адрес ячейки</i>	<i>Текст, вводимый в ячейку</i>	<i>Адрес ячейки</i>	<i>Текст, вводимый в ячейку</i>
<b>B3</b>	Поступ. в банк плат.	<b>E30</b>	КПП
<b>B5</b>	ПЛАТЕЖНОЕ ПОРУЧЕНИЕ №	<b>K17</b>	Сч. №
<b>B9</b>	Сумма	<b>K20</b>	БИК
<b>B10</b>	прописью	<b>K22</b>	Сч. №
<b>B12</b>	ИНН	<b>K25</b>	БИК
<b>B18</b>	Платательщик	<b>K27</b>	Сч. №
<b>B23</b>	Банк плательщика	<b>K30</b>	Сч. №
<b>B28</b>	Банк получателя	<b>K34</b>	Вид оп.
<b>B30</b>	ИНН	<b>K35</b>	Наз. пл.
<b>B36</b>	Получатель	<b>K36</b>	Код
<b>B45</b>	Назначение платежа	<b>O7</b>	Вид платежа
<b>C51</b>	М.П.	<b>P48</b>	Отметки банка
<b>I7</b>	Дата	<b>Q34</b>	Срок плат.
<b>G48</b>	Подписи	<b>Q35</b>	Очер. плат.
<b>K12</b>	Сумма	<b>Q36</b>	Рез.поле
<b>E3</b>	Списано со сч. плат.	<b>U3</b>	401060
<b>E12</b>	КПП		

6. Используя режим рисования линий, вызвав панель «Рисование», проведите пять *горизонтальных* линии минимальной толщины, длиной 35 мм:

- ✓ две между 2 и 3 строкой (столбцы В и С, Е - I);
- ✓ две между 6 и 7 строкой (столбцы I - N и Р - Т);

✓ одну посередине 15 строки от столбца К до столбца W.

7. Выполнив команду меню *Вид/Панели управления/Граница*, проведите линии границ у ячеек:

- ✓ горизонтальные границы внизу от ячеек (диапазонов):  
 B11:W11, B13:J13, B19:M19, B24:W24, K21:M21,  
 K26:M26, B29:M29, B31:J31, K33:W33, K34:M34,  
 K35:M35, Q34:R34, Q35:R35, B37:W37, B38:W38,  
 B46:W46, E50:K50, E53:K53
- ✓ вертикальные границы справа от ячеек (диапазонов):  
 B11:B11, D12:D13, D30:D31, J12:J37, M13:M37,  
 P34:P36, R34:R36, C38, G38, H38, K38, Q38, U38
- ✓ внешние границы вокруг ячеек (диапазонов):  
 U3:W4, W6:W7

8. Разрешите ввод информации только в те ячейки, которые для этого предназначены, а остальные защитите от изменения. Это следует сделать для того, чтобы пользователь не изменил служебной информации и всего вида платежки. Для этого выполните следующие действия:

✓ Установите на панель быстрого доступа две дополнительных кнопки кнопку *Защитить лист* и *Блокировать ячейку*. На рисунке 3.2 это кнопочки с изображением замка.

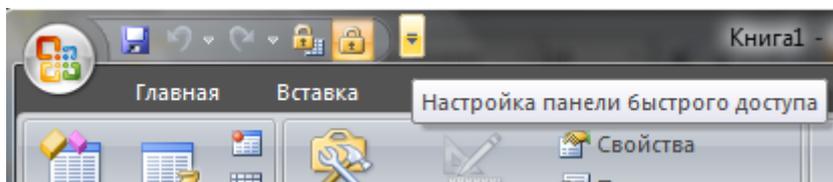


Рисунок 3.2. – Вид панели быстрого доступа.

Для этого, сначала, щелкните по крайней правой кнопке на панели быстрого доступа, выполните команду «*Другие команды*», или щелкните правой кнопкой мыши по «*Ленте*», а после этого активизируйте команду «*Добавить на панель быстрого доступа*». В обоих случаях будет открыто окно «*Параметры Excel*», представленное на рисунке 3.3.

Из левого списка выберите строку «*Настройка*». В левом верхнем раскрывающемся списке «*Выбрать команды из..*» выберите строку «*Вкладка Рецензирование*». После щелчка по кнопке «*Добавить*» выделенная строка с именем команды будет записана в правый список, а после щелчка по кнопке «*ОК*» попадет на панель быстрого доступа.

Аналогично поступите со второй кнопкой. Перечень команд для этого выглядит так *Параметры Excel/Настройка/Выбрать команды из../Вкладка «Главная»/команда «Блокировать ячейку»/Добавить*.



9. Защитите лист. В данном случае, вводить пароль смысла нет. Но при этой операции можно поднять флажки для тех опций которые будут что-то позволять пользователю при работе с заблокированными ячейками, например «Выделение заблокированных ячеек» -см. рисунок 3.4

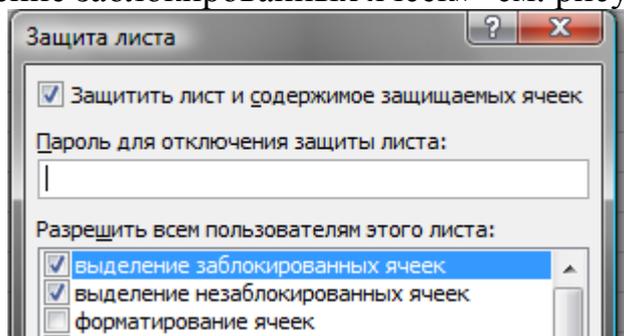


Рисунок 3.4. – Рабочее окно для режима «Защита Листа».

10. Запомните Книгу.

11. Введите необходимый текст и получите платежное поручение, представленное ниже.

Переход от одной ячейки к другой для ввода информации осуществляется с клавиатуры кнопкой табуляции или мышью. В заблокированные ячейки ввод информации невозможен.

## 2. ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Создайте бланк «Расходного кассового ордера».
2. Создайте бланк «Приходного кассового ордера».
3. Создайте бланк «Счета-фактуры».

Разметку бланков при их создании следует производить сначала слева на право, а потом сверху вниз, каждый раз печатая откорректированный вариант и сверяясь с готовым бланком. Ваша попытка создать сразу все окончится неудачей.

## 4. ИССЛЕДОВАНИЕ ВСТРОЕННЫХ ФУНКЦИЙ И ПРОВЕРКА РАБОТЫ ОПЕРАЦИЙ

При работе в этой части пособия рекомендуется составить отчет или конспект. Он должен включать все выполненные задания (1-6), с таблицами, примерами и комментариями по работе рассмотренных функций. В тех случаях, когда это возможно и целесообразно, материал разместите на дискете, остальное законспектируйте с соответствующими выводами.

*Учтите, что информация в "Окне Проверки", "Окне контрольных значений" и др. сбрасывается после закрытия приложения, и сохранить полученные в этом окне результаты на дискете или ином носителе не представляется возможным. Поэтому данные из этого окна конспектируйте или сохраняйте в каком-либо редакторе как рисунок !*

### 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ.

#### 1.1. Основные окна, используемые при отладке программ.

Набор текста программ и изменение его в среде Office происходит в редакторе *Visual Basic for Application (VBA)*. Его вызов может быть произведен из любого приложения Office горячими клавишами  $\langle Alt+F1 \rangle$ , или из вкладки меню «Разработчик», из группы команд «Код», кнопкой «Visual Basic».

Для отладки программ, проверки работы большинства операций и функций в Excel предусмотрена удобная возможность использования ряда окон, которые открываются после запуска редактора.

	<u>Английское название</u>		<u>Русское название</u>
	Immediate Window	Ctrl+G	=Окно "Проверка"
	Local Variables Window		=Окно "Локальные переменные"=
	Watch Window		=Окно "Контрольные значения"=

Окно "*Локальные переменные*" автоматически отображает все объявленные переменные текущей процедуры и их значения.

Окно "*Контрольные значения*" применяется для просмотра значений выражений и переменных. Но для этого необходимо сначала перетащить мышью такие переменные из текста программы в окно.

Окно "*Проверка*" позволяет ввести инструкцию, которая будет немедленно выполнена после нажатия клавиши Enter:

- ✓ при необходимости вывода результатов расчета в этом окне перед выполняемой операцией поставьте слово «**print**» или знак **?**.
- ✓ вывод в окно возможен и при выполнении программ. Значение любой *переменной* из программы может быть выведено на панель «Проверка» методом **Debug.Print переменная**

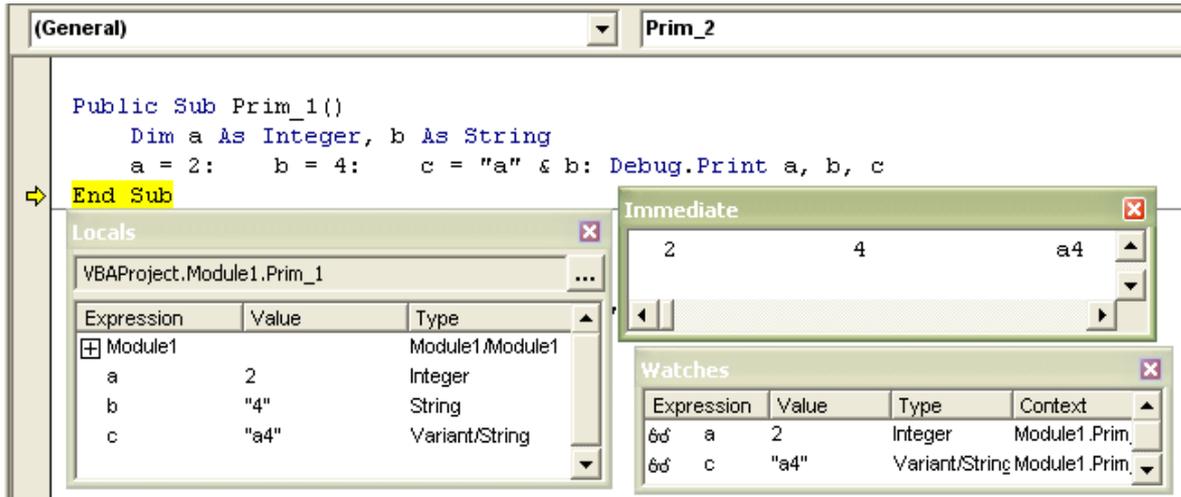


Рисунок 4.1.- Пояснения к заданию №1.

**Задание №1.** Выполните инструкции представленные на рис.4.1 и убедитесь в вышеизложенном:

- ✓ запустите редактор VBA;
- ✓ в окне модуля наберите текст процедуры, представленной на рисунке.4.1.;
- ✓ сделайте активными три указанных окна;
- ✓ щелкнув по имени процедуры, и нажимая после этого кнопку <F8> на клавиатуре, выполните программу в пошаговом режиме, оценивая и объясняя результаты появляющиеся в окнах.

**Задание №2.** Проверьте действия, выполняемые инструкциями (функциями), в соответствии с рисунком 4.2

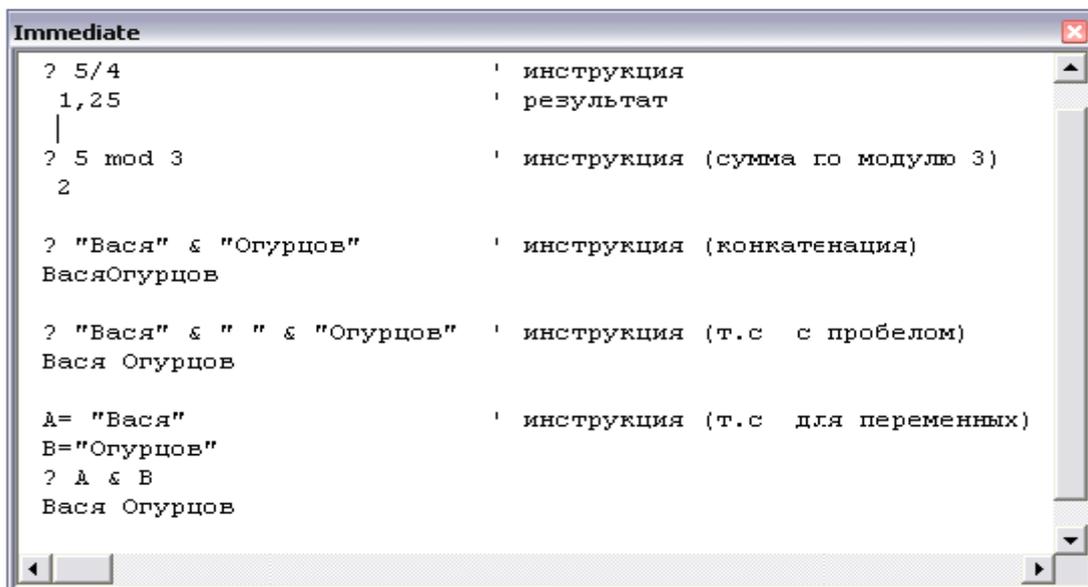


Рисунок 4.2 – Вид результатов для задания №2.

## 1.2 Управление печатью в окне «Immediate» (Проверка) с использованием разделителей

При выводе информации на печать в окне «Immediate» (Проверка) (см. рисунок 4.3) имеют место следующие закономерности:

- ✓ при размещении в строке для печати нескольких элементов, разделенных *запятыми* (,), они будут напечатаны в блоках, изначально заполненных пробелами по 14 символов в блоке.
- ✓ при разделении элементов символом (;), напечатанная строка будет содержать по два пробела между числами: один ведущий, перед числом и второй конечный, после числа. Они появляются в результате преобразования числа в строку оператором вывода на печать.
- ✓ при разделении элементов символом (;) и печати символьных переменных пробелы не ставятся, и все слова на панели будут написаны слитно

**Задание № 3.** Выполните инструкции представленные на рисунке 4.3. При этом, обратите внимание на то, что было сказано выше, а также и то, что:

- ✓ ? и **Print** имеют один смысл как управляющие операторы;
- ✓ строка '1234567890123#12....использована в примере как линейка, указывающая на позицию символов в строке над ней, причем символ # указывает на позицию каждого 14-го символа

```

Immediate
? 5/7, 8/20, 400/10          'разделены через запятую
0,714285714285714          0,4          40
? 5/7; 8/20; 400/10        'разделены через точку с запятой
0,714285714285714 0,4 40

print 1,22222,3333333333,12345678901,99
1          22222          3333333333  12345678901  99
'1234567890123#1234567890123#1234567890123#1234567890123#1234567890123#  'Линейка

print 1;22222;3333333333;12345678901;99
1 22222 3333333333 12345678901 99

print 11,22,33,44
11          22          33          44  'каждый блок 14 символов
print 11;22;33;44
11 22 33 44          'между числами по 2 пробела

print "11", "22", "33", "44"
11          22          33          44
print "11";"22";"33";"44"
11223344
  
```

Рисунок 4.3 – Печать с использованием разделителей

### 1.3. Управление печатью с помощью функции «Format»

В упрощенной форме синтаксис функции формат «*Format*» выглядит так:



#### ***Преобразования числовых величин.***

Первым аргументом функции *Format* является само число, которое необходимо напечатать (в данном случае напечатать на панели «Отладка»).

Вторым аргументом является строка форматирования, определяющая в каком виде это число будет напечатано, и содержащая ряд символов для указания этого вида:

- ⇒ символы (#) - используются в качестве цифровых заполнителей;
- ⇒ символы (, запятая) , указывают на то что каждые 3 символа отделяются запятыми;
- ⇒ символы (. - точка), указывают на положение десятичной точки;
- ⇒ символы (0 - ноль), указывают на позицию обязательного символа.

#### **Задание № 4.**

1. Выполните инструкции представленные на рисунке 4.4. Объясните полученные результаты.

```

a=33.1234 'мжду целыми и дробными десятичная точка !!!
b=0.12345
c=12345
print a, b, c
  33,1234      0,12345      12345

Print format(a, "#.0"), format(a, "#.00"), format(a, "0.000"), format(a, "000.000")
33,1          33,12         33,123         033,123
Print format(b, "#.0"), format(b, "#.00"), format(c, "0.000"), format(c, "#,###.000")
,1            ,12           12345,000      12 345,000
  
```

Рисунок 4.4 – Результаты преобразования числовых величин

2. Заданы следующие числа:  $a=98733,12345$  и  $b=0,0012345$

Задайте такой формат, чтобы напечатать их в следующем виде

а 98733,1 98733,12 98733,123 098733,1234 98 733,1  
98733,12

в ,0 ,001234500

3. Выполните преобразования по *именованным форматам*, представленные на рисунке 4.5.

```

Immediate
? Format (1234.1234567, "General Number") ' Число без выделения разрядов тысяч
1234,1234567
? Format (1234.1234567, "Fixed") ' Как минимум один знак слева и два справа
1234,12 ' от десятичного разделителя
? Format (.1, "Fixed") ' без выделения разрядов тысяч
0,10

? Format (1234.1234567, "Standard") ' Число с выделением разрядов чисел и
1 234,12 ' как минимум одним знаком слева и двумя справа
' от десятичного разделителя
? Format (1234.1234567, "Currency") ' Число с выделением разрядов чисел, с двумя знаками
1 234,12р. ' после десятичного разделителя,
1 234,12р. ' а также символом денежной единицы

```

Рисунок 4.5 - Преобразования с использованием именованных форматов.

### *Преобразования даты и времени по заданным форматам.*

#### Задание № 5.

Выполните преобразования в соответствии с материалом представленным на рисунке 4.6

```

Immediate
? Format (Date, "mm/dd/yy")
11.10.03
? Format (Time, "hh:mm")
02:43
? Format (Now, "mm/dd/yy hh:mm")
11.10.03 02:44

' Именованные форматы
' -----
? Format (Date, "General date") ' дата, в соответствии с установками в панели управления.
10.11.2003
? Format (Date, "Long Date") ' дата в виде, заданном в панели управления
10 Ноябрь 2003 г. ' для полного формата даты.

? Format (Date, "Medium Date") ' для среднего формата даты.
10-ноя-03
? Format (Date, "Short Date") ' для краткого формата даты.
10.11.2003

' АНАЛОГИЧНО для форматов - Long Time, Medium Time, Short Time

```

Рисунок 4.6 – Преобразование даты и времени по заданным форматам.

### ***Создание собственных форматов.***

Созданию собственных форматов было уделено внимание в подразделе 1.2.

Подобные форматы можно создать для строковых переменных. Например, для того, чтобы перевести символы в верхний регистр, следует выполнить такую инструкцию:

```
? Format ("вася", ">")
ВАСЯ
```

Все эти форматы называют - *User-Defined Formats*.- пользовательские форматы

**Задание № 6.** В справочном материале VBA отыщите разделы и примеры посвященные пользовательским форматам. Приведите свои примеры их использования.

Отформатируйте данные выданные преподавателем к указанному им виду.

**Примечание:** функция «Формат» -*Format* - функционирует совершенно одинаково и в программах при форматировании переменных и в при работе с дисковыми файлами.

**1.4. Особые условия.** Если необходимо проверить работоспособность выражения, состоящего из нескольких операторов, то следует записать их в одну строчку и поставить между каждой парой операторов двоеточие - : .

**Задание № 7** Запустите в окне отладки трехстрочный цикл выводящий цифры от 1 до 10. Слово «вывести» замените необходимым для этого оператором.

```
For i=1 to 10
вывести i
Next i
```

## **2. ИССЛЕДОВАНИЕ РАБОТЫ СТРОКОВЫХ ФУНКЦИЙ**

### **Задание № 8 .**

Внимательно просмотрите содержимое таблицы 4.1.

Повторите примеры, в которых выполняют строковые функции, приведенные ниже, после таблицы. Курсивом в примерах выделены комментарии. Поэкспериментируйте с функциями. Перечертите таблицу, и, по мере выполнения функций, заполните столбец комментариев в таблице своими замечаниями.

Таблица № 4.1

№ п.п.	Функция	Описание	Примечание (Заполняем сами)
1	Lcase	Преобразовывает строку в нижний регистр	
2	Ucase	Преобразовывает строку в верхний регистр	
3	Len	Определяет длину строки	
4	Instr	Ищет подстроку	
5	Lset	Выравнивает строку по левому краю	
6	Rset	Выравнивает строку по правому краю	
7	Left	Выделяет левую часть строки	
8	Right	Выделяет правую часть строки	
9	Mid	Выделяет или перемещает подстроку	
10	LTrim	Удаляет ведущие пробелы	
11	RTrim	Удаляет завершающие пробелы	
12	Trim	Удаляет пробелы с двух сторон строки	
13	Asc	Возвращает ASCII код символа	Что такое ASCII код ?
14	Chr	Возвращает символ по ASCII коду	
15	Str	Преобразует число в строку	
16	Val	Преобразует строку в число	
17	StrComp	Сравнивает 2 строки	См. Таблицу №3

### Примеры выполнения некоторых строковых функций.

**A**="Маленький мальчик нашел пулемет"

**B**="Больше в деревне никто не живет"

? **Lcase(A)**

*лы*

маленький мальчик нашел пулемет

*преобразуем в строчные символы*

? **Instr(B,"никто")**

18

*ищем место подстроки "никто"*

? **Left(A,9)**

A

Маленький

*возвращает 9 первых, левых символов переменной*

? **Mid(B,18,5)**      возвращает 5 символов , начиная с 18-го из переменной  
*B*  
 НИКТО

**C="      Hello      "**      переменной *C* присваиваем значение содержа-  
 щее пробелы.

? **"|" & Ltrim( C) & "|"**      удаляет пробелы слева  
 |Hello      |

? **Val("rwq123")**

0

? **Val(" 123")**

123

? **Val(" 123.7fds")**

123,7

*Функция VAL игнорирует любое количество пробелов слева и преобразовывает строку в число до тех пор пока не встретиться символ, который не может быть частью числа .  
 Между соседними цифрами м.б. любое количество пробелов.  
 Дробное число должно иметь десятичную точку в аргументе.*

### Задание № 9

Исследуйте работу функций, приведенных в Таблице № 4.1, которые не были рассмотрены в примерах.

Примечание. Для исследования функций **Lset** и **Rset** сначала напишите процедуру - **qq()**, необходимую для объявления фиксированной длины строки. Это связано с тем, что для этих функций нужно знать, где у строки правый край, а где левый.

Более того, при выполнении процедуры необходимо помнить то, что локальные переменные "умирают" при выходе из подпрограммы, поэтому их значения необходимо анализировать до момента окончания процедуры. Для этого процедуру останавливают до момента ее окончания, используя оператор Stop.

Длину строки задает оператор *Dim*. А так как в окне отладки оператор *Dim* не работает и длину строковой переменной не задать, приходится это делать в «фиктивной» процедуре **qq()**.

**! Заметим**, что эти используемые переменные и процедура пригодятся при исследовании функций логических проверок.

Ниже приведен текст, который следует набрать в модуле редактора VBA.

'Объявления переменных (3 строки ) сделайте в разделе деклараций модуля.

Dim D As String \* 10 ' объявляем D строковой переменной, длиной 10 символов

Dim D01 As String \* 10, D02 As String \* 10

```
Dim D3 As Variant, D4 As Object, DD(10) As Integer, D1 As Date
Sub qq()
```

```
'-----
```

```
    D = "123456789012345"
```

```
    D1 = Day(Now)
```

```
    D3 = Null
```

```
    LSet D01 = "qaz"
```

```
    RSet D02 = "qaz"
```

```
    Stop ' Это замечательный оператор. Он останавливает, но
           не прерывает 'программу. А это позволяет прове-
           рить значения переменных и 'использовать их в ок-
           не отладки..
```

```
End Sub
```

После набора кода программы запустите ее и покажите чему равно **D1**.

Ответе как сделать так, чтобы **D1** было числом ?

### 3. ИССЛЕДОВАНИЕ РАБОТЫ ФУНКЦИЙ из КАТЕГОРИИ «ДАТА И ВРЕМЯ».

#### Задание № 10 .

Повторите инструкции, приведенные на рисунке 4.7.

Перечертите таблицу 4.2.

Исследуйте работу приведенных в ней функций.

Учтите, что функции **Day()**, **Month()**, **Weekday()**, **Year()**, **Hour()**, **Minute()** используют в качестве аргумента последовательный номер даты так, как это показано в примере, на рисунке 4.7.

```
Проверка
? Now
26.08.99 16:44:40

? DateSerial(99,10,18),DateSerial(1999-5,10-3,18+5)
18.10.99      23.07.94
18.10.99      23.07.94
'Использует 3 аргумента: год, месяц,день, причем строго в указанном порядке.
'С аргументами можно производить арифметические действия

? DateValue("1 февраль 1973"),DateValue("13.03.1950")
01.02.73      13.03.50
? DateValue("30 августа"),DateValue("24-авг-99"), DateValue("Сентябрь 1 1999")
30.08.99      24.08.99      01.09.99
' Переводит аргумент-строку в дату. Правильно обрабатывает все допустимые даты

?Day(Now), Day(DateSerial(99,10,18)),Day(DateValue("30 августа"))
26            18            30
```

Рисунок 4.7 – Вид окна отладки.

Таблица 4.2

№ п.п.	Функция	Описание	Примечание (Заполняем сами)
1	Date	Устанавливает или возвращает текущую дату	
2	Time	Устанавливает или возвращает текущее время	
3	Now	Возвращает текущие дату и время	
4	Date-Serial	Преобразовывает в последовательную дату три целых числа: год, месяц, день	
5	Day	Преобразовывает последовательную дату в день месяца	
6	Month	Преобразовывает последовательную дату в месяц года	
7	Weekday	Преобразовывает последовательную дату в день недели	
8	Year	Преобразовывает последовательную дату в год	
9	Hour	Преобразовывает последовательную дату в часы дня	
10	Minute	Преобразовывает последовательную дату в минуты в часе	

**Задание № 11** Исследуйте функции, производящие вычисления над датами:  
*Timer, DateAdd и DateDiff.*

Синтаксис функции *DateDiff*, вычисляющей разницу между *Датой1* и *Датой2*:

**DateDiff(interval, Дата1, Дата2)**

где: первый аргумент - *interval* в *DateAdd* и *DateDiff* может принимать значения:

*Y*ууу – Год, *Q* - Квартал, *m*- Месяц, *ww*- Неделя,  
*H*- Часы, *N*- Минуты, *S*- Секунды.

- ✓ Повторите примеры выполнения некоторых таких функций, приведенных ниже.

```
? Timer          'Возвращает число секунд прошедших после полуночи
61290,47
? Now
26.08.99 16:56:27
?DateAdd("N",15,Now) 'Добавляет или вычитает заданный временной интервал из даты
26.08.99 17:22:02  '1-й арг. - строка,указывающая тип добавляемого временного интервала
                   '2-й арг. - число временных интервалов
                   '3-й арг - дата к которой добавляют (вычитают) временной 2-й арг.
```

- ✓ Подсчитайте сколько секунд и минут осталось до полуночи.
- ✓ Посчитайте сколько секунд, минут, часов, дней, месяцев Вы прожили от рождения до сего момента.
- ✓ Определите каким днем недели будет день вашего рождения через 1 год, 10 лет (используйте функцию DateAdd)

#### 4. ИССЛЕДОВАНИЕ РАБОТЫ ЛОГИЧЕСКИХ ОПЕРАТОРОВ.

**Задание № 12** Исследуйте операторы логического сравнения, приведенные в таблице № 4.3. Пример такого исследования приведен ниже на рисунке 4.8.

```
? 1=1,          1=2,          1>2,          1<2,          2<=2
True           False         False         True          True

? "qaz" >"qaz1",      "qaz"="qaz"
False           True

? StrComp("qaz","qaz",1) , StrComp("qaz","qaz",0)
0               0
|
? StrComp("qaz","wsx",1) , StrComp("qaz","wsx",0)
-1              -1

? "Вася" Like "В*",   "Вася" Like "Ba??", "Вася" Like "Ba??"
True               True           True

? "Вася" Like "Ba[нс]я", "Вася" Like "Ba[лера]?"
True               False
```

Рисунок 4.8 – Пример выполнения логических операторов.

Таблица № 4.3

№ п.п.	Оператор	Описание	Примечание (Заполняем сами)
1	=	Равно	
2	><	Не равно	
3	<	Меньше	
4	>	Больше	
5	<=	Меньше или равно	
6	>=	Больше или равно	
7	Is	Идентично (только для объектов) Возвращает True, если обе сравниваемые переменные ссылаются на один и тот же объект.	
8	Like	Спец.операция позволяющая использовать при сравнении строк подстановочные символы.: <ul style="list-style-type: none"> <li>■ * - любое кол-во символов, ? - любой символ,</li> <li># -любая цифра,</li> <li>■ [список символов] - любой символ из списка,</li> <li>■ [!список символов] - любой символ не из списка</li> </ul>	
9	StrComp	Сравнивает 2 строки. Имеет 3 аргумента. Синтаксис: StrComp(строка1, строка2, код_сравнения). Если строка1>строка2, то возвращает +1, если строка1=строка2, то возвращает 0, если строка1<строка2, то возвращает -1. Цифровой код код_сравнения определяет двоичное (0) или текстовое (1) сравнение.	

## 5. ФУНКЦИИ ЛОГИЧЕСКИХ ПРОВЕРОК

Функции, приведенные в таблице № 4.4 служат для проверки: относится ли выбранная переменная к величинам нужного вам типа и возвращают *True*, если это так, и *False* в противном случае.

**Задание № 13**

✓ Перед исследованиями в окне "**Проверка**" сначала запустите процедуру **qq()**, приведенную выше в задании № 9. Объясните зачем ?

✓ Ответьте: что будет, если в общую область модуля (проекта) вставить оператор *Option Explicit*.?

✓ Проверьте работу функций, представленных в таблице 4.4.

✓

Таблица №4.4

№ п.п.	Функция	Описание	Примечание (Заполняем сами)
1	IsArray()	Возвращает True, если аргумент массив.	-
2	IsDate()	Возвращает True, если аргумент дата.	-
3	IsNull()	Возвращает True, если аргумент Null.	-
4	IsNumeric	Возвращает True, если аргумент число	-
5	IsObject()	Возвращает True, если аргумент объект.	-
6	IsMissing()	Возвращает True, если аргумент опущен.	Проверяет передан ли аргумент процедуре?

Примеры исследований в окне "Проверка" должны иметь вид:

? D

1234567890

? D1

15.01.1900

? IsArray(D), IsArray(DD), IsNull(D3)

False True True

? IsDate(D), IsDate(D1), IsNumeric(DD), IsObject(D4)

False True False True

## 5. РЕДАКТОР VBA.

### 1. ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ. ОБЩИЕ ПОЛОЖЕНИЯ.

Разработка программного обеспечения на VBA для работы с приложениями в Office практически полностью выполняется в редакторе VBA. Для приложений Microsoft Excel, Word, PowerPoint редактор представляет полноценную *интегрированную среду разработки (IDE)*, позволяющую создавать, редактировать, отлаживать, запускать программы, связанные с документами Office

Работа в редакторе предполагает, что Вы создаете *проект* программного продукта, который включает и модули с программным кодом, и модули классов, и пользовательские формы, и отдельные листы и документы и др., то есть все то к чему в рамках данной разработки пользователь имеет доступ.

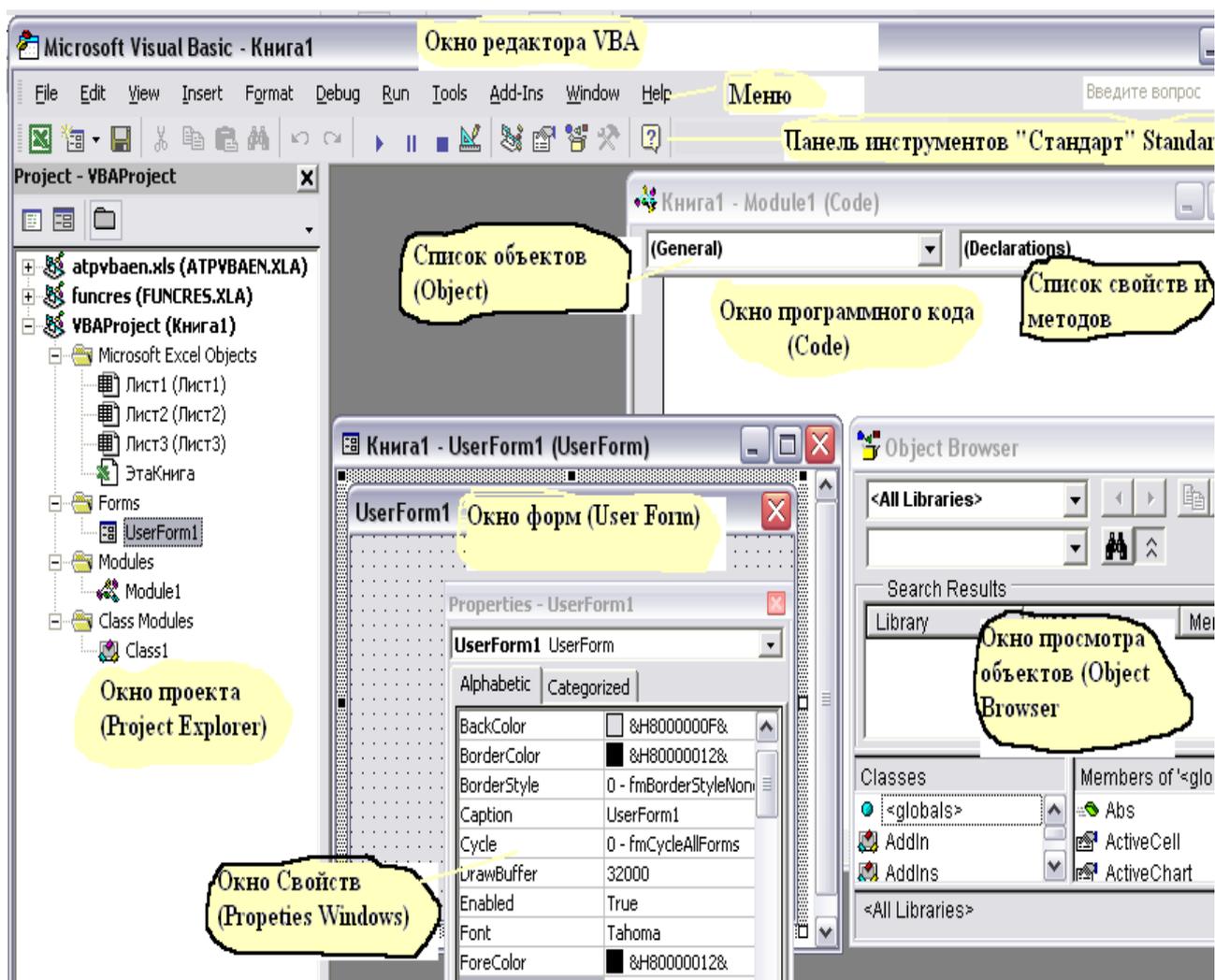


Рисунок 5.1 – Основные окна редактора VBA.

Для вызова редактора VBA следует после запуска основного приложения (например, Excel) выполнить команду меню: *Разработчик/Код/Visual Basic*, или нажать горячие клавиши  $\langle Alt+F11 \rangle$ . В общем случае, редактор VBA включает большое количество окон, панелей инструментов, которые могут быть, при желании, убраны или добавлены в пользовательский интерфейс. Один из его вариантов может иметь вид, представленный на рисунке 5.1.

Для открытия окон используйте соответствующие команды из меню редактора VBA *Вид\Окно проверки (View\Immediate Windows)*, или *Вид\Окно проекта (View\Project Explorer)* и т.д. или горячие клавиши, которые обозначены в меню, рядом с командой. Например, для *открытия окна проекта* -  $\langle Ctrl + R \rangle$ , а *окна свойств* -  $F4$  (при этом в последнем выводятся свойства активного объекта).

Используя команды меню *Вставка (Insert)*, можно добавлять в проект (окно проекта) *Формы (User Form)*, *Модули (Module)*, *Модули класса (Class Module)*, а в окно с программным кодом *процедуры (Procedure)* или *файл (File...)*.

Для активизации (вывода) на экран *окна с программным кодом (окна модуля (Module))* следует дважды щелкнуть мышью в окне проекта по имени того приложения, для которого необходимо создать какую-либо программную единицу. Аналогично можно вызвать окно кода для любой формы или ее элемента управления.

*Примечание.* Правая кнопка мыши, по щелчку выводит контекстно-связанные меню ко всем элементам (окнам, формам, кнопкам и т.д.).

**Задание 1** Научитесь вызывать и закрывать редактор VBA; открывать окна, изменять их размеры, двигать по экрану, открывать панели инструментов; вводить программный код; изменять свойства элементов, используя окно свойств. Запомните указанные горячие клавиши.

## 2. ОКНА РЕДАКТОРА VBA.

### 2.1. Окно проекта ( Project explorer).

Для скрытия окна используйте  кнопку -, или команду *Hide (Скрыть)* из контекстного меню.

Если из этого же меню выполнить команду *Dockable (Закрепить)*, то окно всегда будет располагаться поверх других окон, и вместо 3-х кнопок в его правом верхнем углу останется одна.

Используя команду *Remove...(Удалить)* из контекстного меню, можно удалять формы, классы и модули.

Щелкнув по объекту в окне проекта вы выделяете активизируете под его и его окна, а двойной щелчок по объекту из окна проекта вызывает, как правило, окно модуля (программного кода) с набором процедур написанных для этого объекта.

Для вывода информации ветвей дерева в окне проекта щелкните по символу "+" около имени ветви. Щелчок по символу " – " скрывает информацию ветвей.

## 2.2. Окно модуля.

Это окно используется для создания текста любой программы. Его особенностью является присутствие двух раскрывающихся списков под строкой заголовка окна. В левом выводятся все объекты модуля (на рисунок 5.2). В данном случае из них выбран Worksheet. В правом списке выводится набор процедур (событий), связанных с данным объектом. При щелчке мышью по имени этой процедуры в окне модуля появляются ее первая и последняя строки (такая процедура называется - пустой процедурой), и вам остается только написать тело процедуры.

Редактирование текста программного кода происходит точно так же, как и в любом текстовом редакторе.

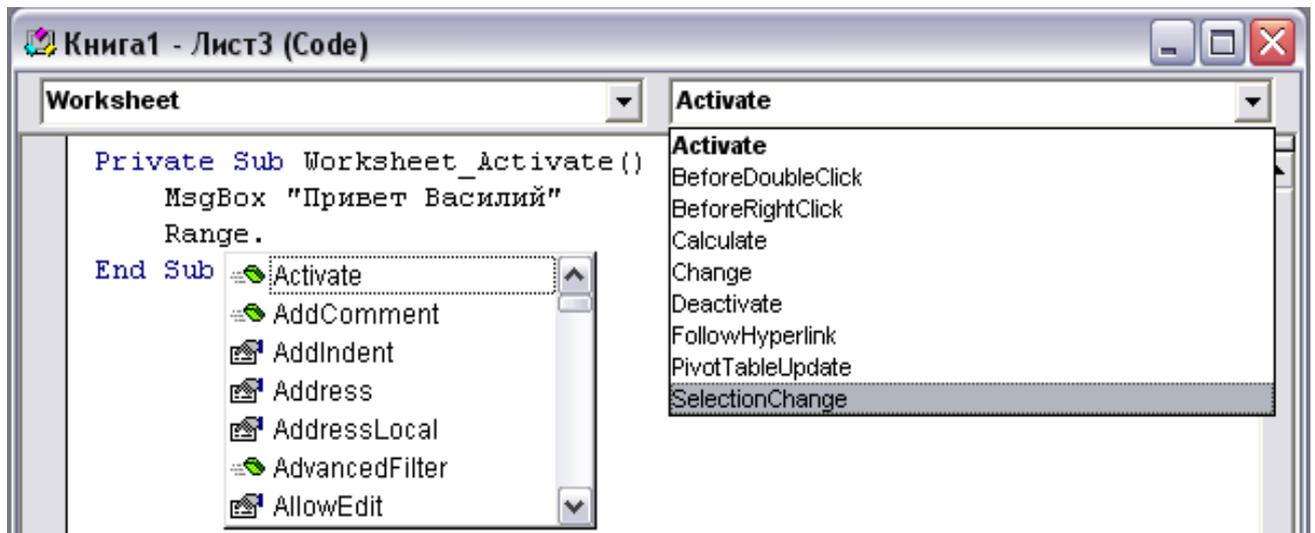


Рисунок 5.2 – Вид окна кода (модуля)

*Процесс работы по написанию кода в VBA автоматизирован.*

✓ Добавления своих пустых процедур и функций происходит по команде меню *Вставка\Процедура(Insert Procedure...)*. При этом, используя появившееся рабочее окно, можно задать необходимые параметры процедур и функций.

✓ При описании переменных (например переменной *qq*), с использованием конструкций вида - *Dim qq As*, редактор предложит список их допустимых типов сразу после набора слова *As*.

✓ В том случае, если нужно вставить в текст программы метод или свойство, можно обратиться к команде основного меню редактора - *Правка\Список Свойств\Методов (Edit\List Properties/Methods)*, или к команде контекстного меню окна кода (щелкните правой кнопкой мыши в любом его месте) - *Список Свойств\Методов (List Properties/Methods)*.

✓ При вводе имени объекта и точки после него редактор отобразит список его методов и свойств, так, как это отображено на рисунке.5.2 для объекта Range. Указанный режим будет действовать в том случае, если в диалоговом окне *Параметры* (команда вызова из меню – *Сервис\Параметры\Редактор (Tools\Options...\Editor)*) предварительно был поднят флажок *Список компонентов (Auto List Members)*.

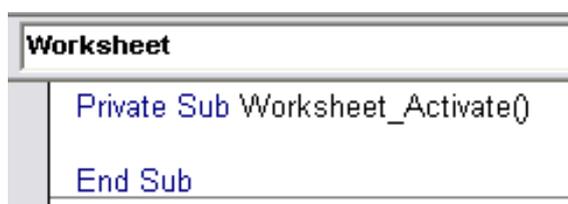
✓ Команда меню *View\Просмотр объектов(View\Object Browser)* или горячая клавиша **F2** вызывают Окно просмотра объектов (см. рисунок 5.1). С помощью этого окна можно определить к какой из библиотек принадлежит объект, посмотреть его свойства и методы, уточнить константы.

**Проанализируем**, что же можно узнать из информации на рисунке 5.2.

Во-первых, код программы создается для объекта «Лист3», который вложен в объект «Книга1».. Об этом сообщает надпись в заголовке окна кода -«Книга1 – Лист3(Code)». В терминах VBA на объект – рабочий лист указывает и имя выбранного объекта в левом раскрывающемся списке – *Worksheet*.

Во-вторых, в правом раскрывающемся списке, который на рисунке открыт и содержит методы и свойства объекта *Worksheet*, выбран метод *Activate*.

В третьих, как только такой выбор будет сделан, в окне кода редактор сделает заготовку для программы обработки события – активизировать Лист3 – первую и последнюю инструкции (рисунок 5.3).



```
Worksheet
Private Sub Worksheet_Activate()
End Sub
```

Рисунок 5.3 – Пустая процедура

Осталось заполнить инструкциями тело процедуры и все.

Первая команда выводит на экран окно вывода с надписью «Привет Василий»

Продолжим вторую команду. Ее нужно несколько исправить. В том виде как она представлена конкретная ячейка рабочего листа, на которую должен ссылаться объект Range, не определена, не определено и свойство объекта Range. Сделаем так, чтобы в ячейку «A3» рабочего листа записать информацию. Запишем команду так:

`Range("A3").Value="1234567"`

**Задание №2** Наберите текст рассмотренной выше процедуры, исправив второй оператор. Перейдите на первый или второй рабочий лист. Активизируйте Лист3. Проанализируйте результат.

**Задание №3** А) Наберите текст процедуры, приведенный на рисунке 5.4.

```
Public Sub Prim_1()
    Dim qq As String
    qq = "Привет !"
    MsgBox "Hello"
    MsgBox qq
    Worksheets("Лист1").Range("A1").Value =
        Worksheets.Parent.Name
    qq = Worksheets.Parent.Name
    MsgBox qq
    ActiveWorkbook.Worksheets(1).Select
    ActiveSheet.Range("A2").Value = Worksheets.Count
End Sub
```

Рисунок 5.4 - Вариант рабочей программы для задания №3.

Набирая текст, обратите внимание на максимально возможную автоматизацию ввода текста программы. Ответьте где здесь объекты, свойства методы, что они обозначают и (или) выполняют.

**В)** Запустите процедуру на выполнение. Для этого курсор мыши поставьте на какой-нибудь оператор программы и выполните команду меню *Run/Run Macro* или щелкните на функциональную клавишу F5 на клавиатуре.

**С)** Попробуйте выполнить программу в пошаговом режиме. В этом случае система будет ждать команду на выполнение очередного оператора. Такой командой является инструкция меню редактора VBA *Debug/StepInto* или щелчок по клавише F8 на клавиатуре.

Если при выполнении программы необходимо уточнить значение какой-нибудь переменной поставьте курсор мыши на эту переменную в тексте программы или воспользуйтесь окнами просмотра или отладки..

В том случае, если требуется закончить программу, не дожидаясь выполнения всех операторов, т.е. прервать программу, щелкните мышью по кнопке **Reset**



**Д)** Поэкспериментируйте !

Предложите свои варианты. Попробуйте все то, о чем говорилось выше.

### 2.3. Окно форм и окно свойств.

В проект можно вставить любое количество форм. Каждая форма вставляется командой меню редактора VBA *Вставка\Форма (Insert\UserForm)* Отдельные элементы управления: кнопка, поле ввода и другие, вставляются в форму перетаскиванием их с панели инструментов (*ToolBox*).

Каждый элемент, размещенный на форме, обладает свойствами, отображаемые в окне свойств, которые могут быть при необходимости изменены. Окно свойств (*Properties Window*) открывается командой меню *Вид\Окно свойств (View\ Properties Window)* или горячей клавишей *F4*.

Изменение элементов в размерах и их перемещение в пределах формы производится обычным для Window образом (активизируйте и тащите или изменяйте мышью). Для выделения нескольких элементов одновременно щелкните последовательно по ним мышью, удерживая нажатой кнопку *Shift*.

Выделяя несколько элементов можно добиться желаемого размещения с помощью команд из меню, сгруппированных под общей командой *Формат*.

**Задание №4** А) Разместите на форме 4 кнопки. Покажите, как работают команды из пункта меню Формат. Измените свойства установленных элементов, используя окно свойств. Позже мы вернемся к этим свойствам, поэтому пока измените интуитивно понятные свойства: имя, цвет и т.д.

С каждым объектом связывают код. Причем, для обработки каждого события, связанного с конкретным объектом, это своя оригинальная программа.

Вызов окна модуля для элементов формы производится двойным щелчком мыши по форме, или при нажатии горячей клавиши *F7*. Выбор объекта и события для создания процедур обработки событий производится так, как это было рассмотрено выше.

**В)** Попробуйте написать для нескольких кнопок программы по обработке события –щелкнуть мышью по кнопке – *CommandButton1\_Click* (здесь цифра один – это номер кнопки), например:

- щелчок по первой кнопке – должен изменить надпись на третьей кнопке (свойство –*Caption*);

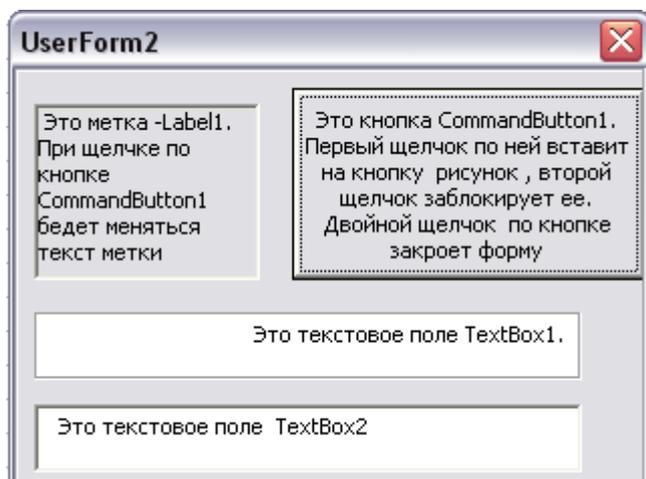
- - щелчок по второй кнопке должен информацию, помещенную в ячейке «A4» первого листа записать в ячейку «B2» второго листа.

## 3. СОЗДАНИЕ ПРОСТЕЙШИХ ПРОГРАММ И ЭКРАННЫХ ФОРМ.

**Задание №5.** (см.рисунок 5.5)

- ✓ Создайте форму с двумя текстовыми полями (полями ввода), одной меткой и кнопкой.

- ✓ Отформатируйте форму, и ее элементы так, как Вам хочется. На кнопку поместите рисунок.
- ✓ С событиями – *изменение текстового поля 2* и щелчок по кнопке свяжите процедуры, текст которых приведен ниже (см. рисунок 5.5).
- ✓ Для запуска формы, для ее активизации – щелкните по любой точке формы мышью, после чего, нажмите кнопку **F5** на клавиатуре.



- ✓ Введите что-нибудь в текстовые поля.
- ✓ Пощелкайте мышью по кнопке.
- ✓ Ответьте и проверьте, что будет, если вместо ключевого слова **Static** ввести **Dim**.
- ✓ Сделайте выводы и обоснуйте ответы.
- ✓ Попробуйте создать свои варианты формы.

Рисунок 5.5 - Пример для выполнения задания №5

```

CommandButton1  DbfClick
Private Sub CommandButton1_Click()
    Static j As Integer
    j = j + 1
    If j = 1 Then
        'Изменяем программно рисунок на кнопке и текст метки
        CommandButton1.Picture = LoadPicture("D:\MSOffice\Clipart\Cat.wmf")
        Label1.Caption = " Вы нажали кнопку 1 раз, нажмите еще"
    Else
        CommandButton1.Enabled = False
        Label1.Caption = " Привет коллега ! Теперь кнопка заблокирована."
    End If
End Sub

Private Sub CommandButton1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    Unload Me          'Закрываем форму
End Sub

Private Sub TextBox2_Change()
    Static i As Integer
    i = i + 1
    TextBox1.Value = " Во втором текстовом поле - изменение № " & CStr(i)
End Sub

```

Рисунок 5.6 – Текст программы для задания №5

#### 4. ЗАДАНИЕ ПАРАМЕТРОВ РЕДАКТОРА.

Для того, чтобы задать параметры редактора необходимо выполнить команду *Сервис\Параметры(Tools\Options..)* и в появившемся окне диалога, на соответствующих вкладках установить их. Помимо явно понятных параметров, отметим следующие:

##### Вкладка редактор:

- ◆ *Проверка синтаксиса (Auto syntax Check)*.
- ◆ *Явное описание переменных(Require VariableDeclaration)* – вставляет оператор *Option Explicit*, требующий явного описания переменных, во вновь создаваемом проекте.
- ◆ *Список компонентов (Auto List Members)* – автоматически отображает список данных, логически завершающих инструкцию, расположенных вместе ее вставки.
- ◆ *Краткие сведения (Auto Quick Inform)* – после ввода имени процедуры отобразит сведения о них и их параметрах.
- ◆ *Подсказки значений данных (Auto Data Trips)* – при установки курсора на переменную выводится ее значение. Работает только в режиме прерывания (отладки программ).
- ◆ *Перетаскивание текста(Drag and Drop Text Editing)* – определяет, возможно ли мышью перемещать и копировать выделенные куски текста в программах.
- ◆ *Просмотр всего модуля (Default to Fill Module View)* – определяет можно ли просматривать процедуры в модуле все сразу или по одной.

##### Вкладка общие(General) :

- ◆ *Всплывающие подсказки (Show ToolTips)*– возникают около кнопок панелей инструментов.

##### Задание №6

- ✓ Создайте новый проект.
- ✓ Получите остальную информацию по заданию параметров редактора щелкнув по кнопке *Справка*, которая находится на каждой вкладке.
- ✓ Попробуйте изменить установки. Проверьте, что изменилось.
- ✓ Используя команду *Сервис\Свойства VBA Project\ Защита (Tools\VBAProject Properties\Protection)* попробуйте защитить свой проект от несанкционированного вмешательства.

**Задание №7**

Ниже представлен вариант программы, состоящий из 4-х процедур. Основная процедура- main запускает в соответствующем порядке 3 остальные процедуры.

Требуется заполнить таблицу, представленную справа, поместив в нее значения переменных- a, b, c, те, какие они будут иметь при каждом выходе из процедур.

После заполнения таблицы, покажите ее преподавателю. После этого наберите текст программы на ПК в окне программного кода и, запустите main на выполнение. В пошаговом режиме проследите, как изменяются значения переменных и их соответствие тем значениям, которые записаны вами в таблице. Объясните все несоответствия, если они есть.

(General)				
Public a As Integer				
Dim b As Integer	Object			
Sub qq1()				
Static c As Integer				
a = a + 1: b = b + 2: c = c + 3				
End Sub				
Sub qq2()				
Dim b As Integer				
a = a + 3: b = b + 4: c = c + 5				
End Sub				
Sub qq3()				
Static a As Integer				
Dim c As Integer				
a = a + 7: b = b + 6: c = c + 9				
End Sub				
Sub main()				
a = 1: b = 2: c = 3				
qq1: qq2: qq1: qq3: qq3: qq2: qq1				
c = a + b + c				
End Sub				

Процедура	Значения переменных при выходе из процедуры		
	a	b	c
main	1	2	3
qq1			
qq2			
qq1			
qq3			
qq3			
qq2			
qq1			
main			

Рисунок 5.7 – Программа для исследования области видимости переменных.

## 6. СОЗДАНИЕ ФУНКЦИЙ, ПРОЦЕДУР И ИХ ОТЛАДКА. ПРИМЕНЕНИЕ ЦИКЛОВ И ПОВТОРЯЕМЫХ СТРУКТУР

### 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ.

Для создания и отладки процедур и функций в Excel, а также других приложениях используют редактор VBA. Он имеет свои панели инструментов и меню отличные тех, которые находятся в основном приложении. Вызовите редактор VBA и убедитесь в этом. Многие основные команды повторяются в головном меню и панелях инструментов, и Вы можете использовать их так, как удобнее.

Основные из тех команд, которые Вам понадобятся при отладке и запуске процедур представлены на рисунке 6.1

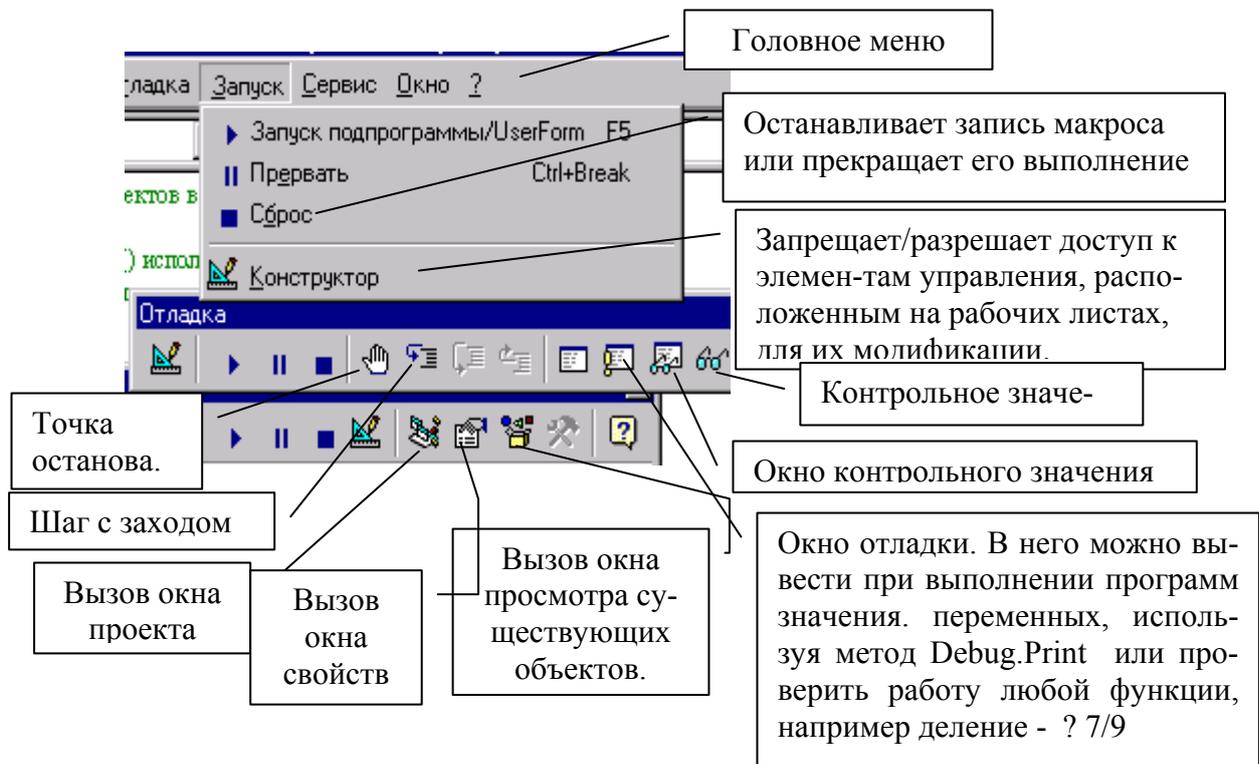


Рисунок 6.1 - Вид меню и панели «Отладка» редактора VBA.

Кнопки "**Запуск программы**" и "**Шаг с заходом**" вызывают выполнение процедуры, причем, во втором случае в пошаговом режиме - все команды данного макроса будут выполняться по очереди. Для этого режима удобнее использовать горячую клавишу **F8**.

Кнопка "**Точка останова**" определяет место в процедуре, где необходимо остановить ее выполнение. Повторный щелчок по кнопке выключает данный режим.

Кнопка "**Контрольное значение**" позволяет вывести имена переменных и их значения в окно "**Контрольного значения**". Перед тем, как установить режим просмотра, перед щелчком по этой кнопке, выделите в исходном тексте процедуры, переменную, информацию о которой Вы хотите получить.

## 2. ЗАПУСК И ОТЛАДКА ПРЦЕДУР.

**Задание №1** Внимательно ознакомьтесь с текстом исходной программы.

Предложенная процедура использует вычисляемый цикл *-For-Next* и заполняет текущий выбор (выделенную область ячеек) случайными числами. Ее исходный текст с подробными комментариями приведен ниже.

1. Наберите текст этой программы в окне модуля (оно появится после выполнения команды меню "**Вставка/Модуль**")

Option Explicit

' Запись случайных чисел в текущий выбор (выделенную область ячеек).

'=====

' Метод Rows возвращает набор всех строк в текущем выборе.

' Метод Columns возвращает набор всех столбцов в текущем выборе.

' Свойство Count возвращает количество объектов в текущем выборе.

'

' Метод Cells, свойство Value и функция Rnd() используются

' для записи случайного числа в текущую ячейку

'

Sub QQ\_Rand()

Dim numRows As Integer, numCols As Integer

Dim theRow As Integer, theCol As Integer

'определение размера текущего набора

numRows = Selection.Rows.Count ' количество строк

numCols = Selection.Columns.Count ' количество столбцов

Randomize ' инициализация генератора случайных чисел.

' Rnd() - функция возвращающая

случ.величину.

For theRow = 1 To numRows ' цикл заполнения строк

For theCol = 1 To numCols ' цикл заполнения столбцов

Selection.Cells(theRow, theCol).Value = Rnd

Next theCol

Next theRow

End Sub

2. Запустите процедуру **QQ\_Rand** и на рабочем листе "Лист1" получите картинку, изображенную на рисунке 6.2. Для этого существует несколько вариантов. Рассмотрим их.

	A	B	C	D	E
1					
2		0,45671	0,833824	0,037694	
3		0,83691	0,894278	0,294622	
4		0,563746	0,783413	0,730796	
5					

Рисунок 6.2 – Результат работы программы QQ\_Rand.

Первый вариант. Активизируйте рабочий лист "Лист1", выделите область ячеек, которую хотите заполнить случайными числами, и выполните команду меню: **Сервис/Макрос/Макросы...**. В появившемся диалоговом окне "Макрос" выберите процедуру **QQ\_Rand** и щелкните по командной кнопке "**Выполнить**".

Второй вариант. Вызовите панель инструментов "**Visual Basic**" (команда меню - **Вид/Панели инструментов/Visual Basic**), и щелкните по кнопке "**Выполнить макрос**". Находясь на рабочем листе и выделите необходимую область. Остальное аналогично.

Третий вариант. Выполните ту же процедуру, но в режиме выполнения операторов по одному -*по шагам*..

Для этого, находясь в "**Окне Модуля**" щелкните по какой-либо команде процедуры **QQ\_Rand**, а потом по  кнопке или клавише **F8** на клавиатуре.

Каждый щелчок по кнопке "**Шаг с заходом**" будет приводить к выполнению очередной команды макроса.

Получите в "**Окне контрольных значений**" информацию о переменных процедуры. Добейтесь того, чтобы это окно имело вид, представленный на рисунке 6.3.

Expression	Value	Type	Context
Rnd	0,8195155	Single	Module1.QQ_Rand
numCols	4	Integer	Module1.QQ_Rand
numRows	5	Integer	Module1.QQ_Rand
theCol	2	Integer	Module1.QQ_Rand
theRow	1	Integer	Module1.QQ_Rand

Рисунок 6.3. - Вид окна контрольных значений.

Выполните последовательно операторы процедуры, щелкая каждый раз по кнопке "*Шаг с заходом*". При этом контролируйте изменения значений переменных и ход выполнения макроса.

### 3. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ

#### 3.1. Определения и особенности создания пользовательских функций.

Все программные единицы делят на процедуры и функции. Функции имеют следующий синтаксис:

```
[Public | Private] [Static] Function Имя_функции [(Список_аргументов)] [As Tun]  
    Тело функции  
End Function
```

Ключевые слова **Function**, **End Function** и *Имя\_функции* являются обязательными параметрами.

*Тело функции* может включать любые допустимые операторы VBA и, в частности, оператор Exit Function, прекращающий выполнение функции и передачу управления вызываемой процедуре. Здесь же *Имя\_функции* получает значение, которое эта функция и будет возвращать.

Описывая функцию можно задать ее *Tun*; сделать ее статической, используя описатель **Static**, указывая на то, что переменные между вызовами функции сохраняются.

Функция описанная как **Private** является доступной в программных единицах текущего модуля, как **Public** – всех модулей. Созданные таким образом функции можно вызывать из любых других процедур и функций. *Но если требуется создать пользовательскую функцию, то необходимо:*

- ✓ описать функцию как **Public**;
- ✓ программный код для функции поместить в окне модуля (который, если он уже создан, можно активизировать из **Окна проекта** либо создать командой меню редактора VBA – **Insert/Module**)

**Задание №2** Учитывая сказанное, создайте пользовательскую функцию так, как это представлено на рисунке 6.4.

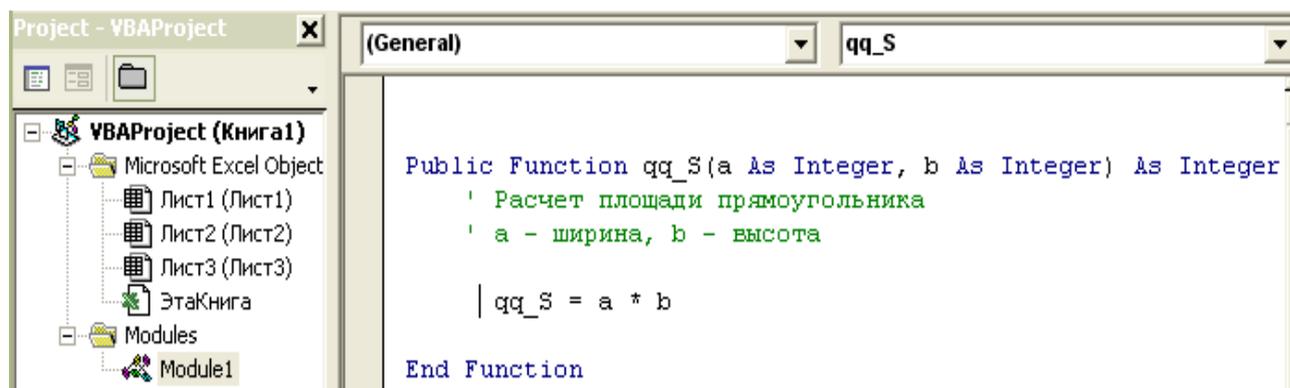


Рисунок 6.4- Окно проекта и модуля для пользовательской программы.

При желании для написанной функции можно добавить комментарии. Для этого, находясь в редакторе VBA, выполните команду меню **Вид/Просмотр объектов (View/Object Browser)**. В появившемся окне, в раскрывающемся списке выбора *Проекта/Библиотек* активизируйте строку **VBAProject** - см. рисунок 6.5, ниже.

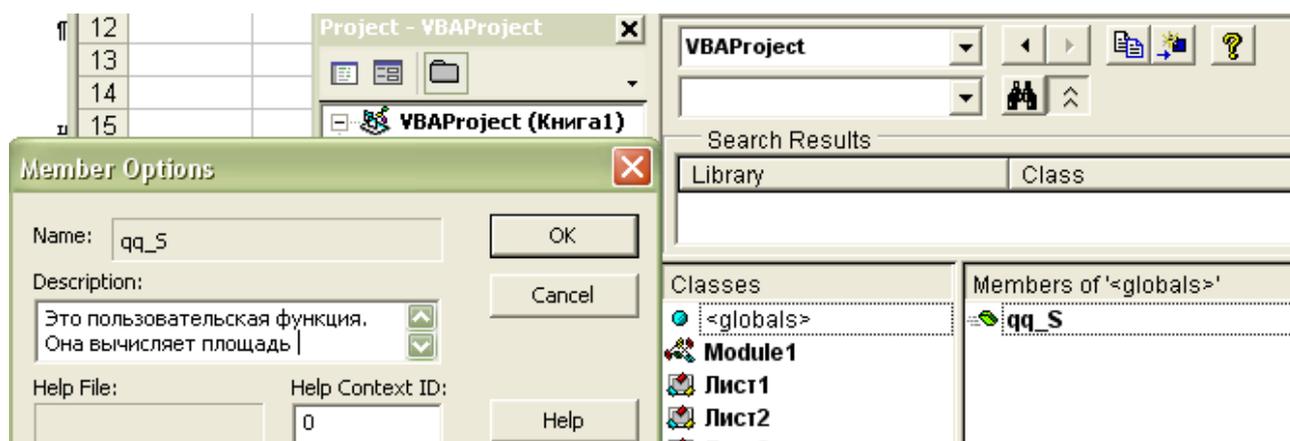


Рисунок 6.5 – Добавление комментариев в пользовательскую функцию.

Щелкните правой кнопкой мыши по имени **qq\_S** вашей пользовательской функции в окне "**Компонент '<глобальные>' (Members of '<globals>')**", и выберите в контекстном меню команду "**Свойства (Properties)**", откройте окно "**Параметры компонента (Member Options)**". В этом окне в поле "**Описание (Description)**" введите поясняющий текст к Вашей функции.

Все созданные таким образом *функции* будут записаны в категорию "**Пользовательских функций**" или "**Определенных пользователем**". После этого **Мастер функций** будет работать с ними так, как он это делает со всеми остальными, встроенными функциями.

На рабочем листе, используя мастер функций, введите написанную функцию в какую-нибудь ячейку. В рабочем окне мастера появится и поясняющий текст, как на рисунке 6.6.

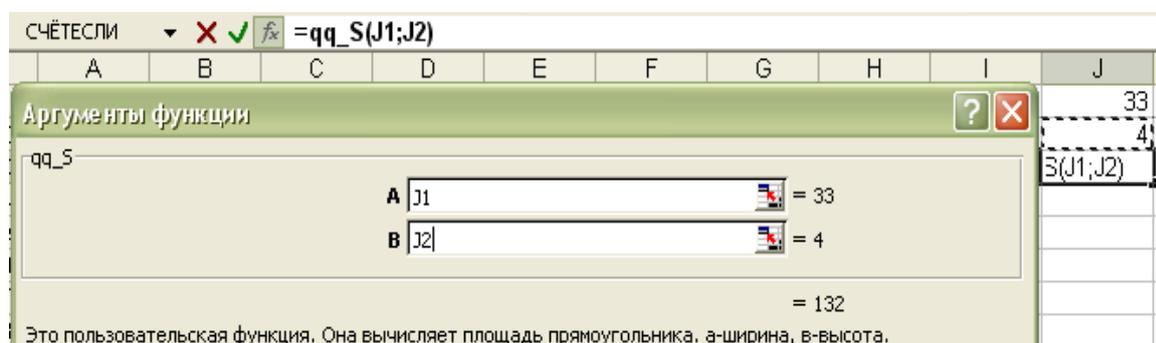


Рисунок 6.6 – Запуск пользовательской функции.

### 3.2. Вызов пользовательских функций для использования в формулах рабочего листа.

**А. Самый простой способ** вызова предполагает использование мастера функций. Он рассмотрен в задании №2, выше. Категория «*Пользовательских функций*» будет доступна мастеру всякий раз, когда открыт файл (книга), в котором были созданы пользовательские функции. Если книгу с пользовательскими функциями закрыть, то в ячейках оставшихся открытых книг, там где в формулах используются эти функции будет возвращено ошибочное значение *#ССЫЛКА*.

Для доступа к функциям всех пользователей запишите свой файл с функциями в каталог **XLSTART**. В этом случае при запуске Excel функции автоматически станут доступны для всех.

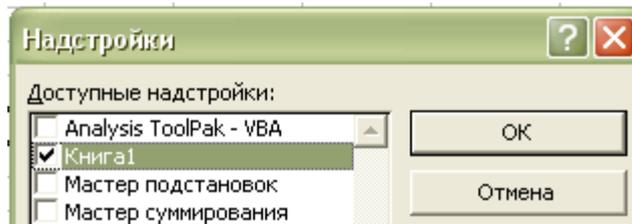
#### Задание №3

- ✓ Попробуйте выполнить созданную пользовательскую функцию из той книги, в которой ее нет.
- ✓ Запишите книгу с созданной пользовательской функцией в каталог **XLSTART**. Закройте Excel. Откройте Excel. Создайте новую книгу и вызовите из нее пользовательскую функцию.

**В. Другой способ** вызова предполагает создание надстройки.

Для этого сохраните файл, в котором находится пользовательская функция используя файловый фильтр: *Надстройка Microsoft Excel (\*.xla)*, например, «*Книга1.xla*», в каталоге *AddIns*.

При работе с новой книгой загрузите надстройку командой меню *Сервис/Надстройки...*, и поднимите флажок около необходимой, в данном случае «Книга1».



После этого вызовите пользовательскую функцию, используя мастер функций обычным образом.

#### Задание №4

Создайте собственную пользовательскую функцию для расчета подоходного налога.

$$\text{Подох\_налог} = (\text{Зар\_плата} - \text{МРОТ} * (\text{Кол\_иждивенцев} + 1) - 1\% * \text{Зар\_плата}) * 12\%$$

Создайте свою надстройку. Покажите, что в новой книге можно загрузив созданную надстройку вызывать и работать с данной пользовательской функцией.

#### Задание №5

Создайте пользовательскую функцию:

***Function Translat\_MesToNum(anMes As String) As Integer***

которая по имени месяца (например, Март) возвращает его номер (в данном случае это - 3).

Используйте структуру Select Case. Проверьте ее работоспособность.

### **3.3. Аргументы пользовательских функций .**

Во всех созданных в данной работе функциях были использованы аргументы различного типа в количестве от одного до нескольких. Тем не менее, следует иметь в виду то, что:

- ✓ аргументы могут отсутствовать;
- ✓ аргументы могут быть обязательными (как в примерах выше), так и необязательными. Перед необязательными следует вставлять ключевое слово ***Optional***.
- ✓ количество аргументов может быть неопределенным. В этом случае перед именем аргумента нужно вставить ключевое слово ***ParamArray***.

### Задание № 6

На рисунке 6.7 приведен текст пользовательской функции без аргументов и пример ее использования.

Создайте свою пользовательскую функцию, которая не использует аргументов, и покажите, что она работоспособна.

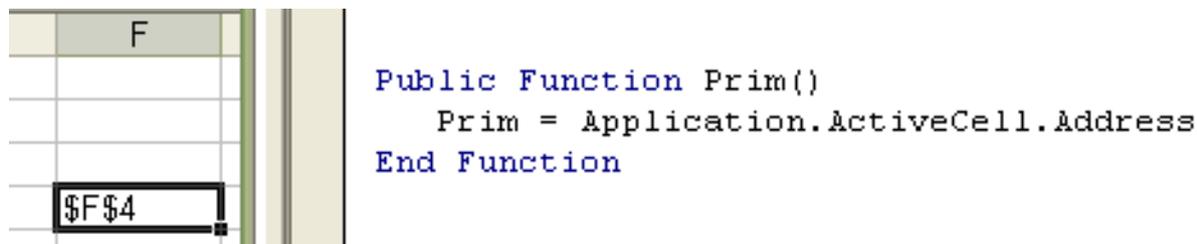


Рисунок 6.7 – Пользовательская функция без аргументов.

### Задание № 7

Создайте функцию с необязательным параметром. В том случае, если этот параметр не получит никакого значения функция должна возвращать одно значение (например имя Катерина), а если параметр задан – то другое значение (другое имя).

### Задание № 8

1. Найдите ошибку, отладьте и запустите функцию, осуществляющую поиск в массиве ячеек на листе (таблице) и возвращающую первый встретившийся ей положительный элемент. В том случае, если такого элемента нет - функция сообщает об этом и возвращает ноль.

Текст функции приведен ниже, на рисунке 6.8. В ней используется логически прерываемый цикл Do-Loop, а также элементы работы с массивами.

2. Добейтесь того, чтобы функция работала как пользовательская.
3. Составьте подпрограмму-процедуру и убедитесь, что функция работает и в этом случае. Искомый результат выведите в окно отладки, массив данных задайте в виде последовательности чисел.

```

Function qwe(theArray() As Variant) As Single
=====
Dim j As Integer, Value As Single
j = LBound(theArray()) - 1
Do
    j = j + 1
    If j > UBound(theArray()) Then
        MsgBox "Такого числа в выбранном массиве нет"
        Value = 0
        Exit Do
    End If
    Value = theArray(j)
Loop Until Value > 0

qwe = Value
End Function

```

Рисунок 6.8 – Программа с ошибками.

### 3.4.Использование функций рабочего листа в процедурах и функциях VBA.

Полный перечень таких функций можно получить открыв в интернете [List of Worksheet Functions Available to Visual Basic \(список функций раб.листа доступных для VB\)](#) или открыв в справке по «Excel Список функций рабочего листа (по категориям)». Все эти функции можно использовать, если при обращении к ним перед именем функции вставить *WorksheetFunction* и точку.

Для примера была выбрана функция *Large(массив, i)*. Функция возвращает *i-е* наибольшее значение из *массива* или *интервала данных* (диапазона рабочих ячеек). На рисунке 6.9 она использована для получения 2-х максимальных значения, хранящихся в пяти выделенных ячейках рабочего листа.

Очевидно, что если массив содержит *N* элементов, то *Large(массив,N)* вернет наименьшее значение.

Рисунок 6.9 – Пример использования функции рабочего листа.

**Задание № 9**

1. Повторите пример, приведенный на рисунок 6.9.
2. Напишите функцию, которая бы возвращала сумму двух минимальных значений массива и двух максимальных. Массив представлен набором выделенных ячеек Excel. В данной задаче нужно определить и размерность массива.
3. Напишите функцию, которая использует в своем теле любую другую функцию рабочего листа. Проверьте ее работоспособность.

**4 СОЗДАНИЕ ПРОЦЕДУР****Задание № 10**

1. Создайте процедуру название и формальные параметры которой представлены ниже.

***Sub Zam\_Simvol ( Sim1 As String; Sim2 As String; \_  
Sim3 As String; Stroka As Variant)***

---

' Sim1 - число, Sim2 - месяц, Sim3 - год  
' из строки Stroka : 25 января 1998 г.

Указанная процедура должна из строки, в которой содержится дата, представленная в формате: **25 января 1998 г.**, выделить и вернуть три других строки. Одну с числом (в данном случае - **25**), другую с месяцем (**января**) и третью с двумя последними цифрами года (**98**). Причем, число пробелов между словами и цифрами в исходной строке, а также перед первым числом и после последнего может быть любым.

2. Проверьте созданную программную единицу. Для этого напишите проверочную процедуру, которая бы из первой ячейки считывала бы исходную строку, а во 2-ю, 3-ю, 4-ю ячейки, и записывала бы результаты, возвращаемые проверяемой процедурой (во 2-ю - число, в 3-ю месяц, в 4-ю год). Сами ячейки и лист можете выбрать любыми.

**Задание № 11** Создайте процедуру для замены любого количества пробелов между символами одним символом табуляции.

Вместо символа табуляции используйте константу **vbTab**.

Для замены используйте встроенную функцию **Replace**.

Создайте и проверьте процедуру для случаев ее использования в *Excel* и *Word*.

**Задание № 12** Используя строковую функцию *Replace*, создайте процедуру для удаления вхождений заданной подстроки из строки.

Создайте и проверьте процедуру для случаев ее использования в *Excel* и *Word*.

**ЛИТЕРАТУРА**

1. Вильям Дж.Орвис, Visual Basic for Application на примерах. - М.: Бинном, 1995. -512 с.
2. К.Соломон. Microsoft Office 97: разработка приложений - СПб.: - СПб,1998. -560 с
3. Камминг Стив. VBA для чайников , 3-издание.:Пер. с англ. - М.: Издательский дом "Вильямс", 2001. - 448 с.
4. Биллиг В.А.. Средства разработки VBA- программиста. Офисное программирование. Т.1. -М.: Издательско-торговый дом "Русская редакция", 2001. - 480 с.
5. Биллиг В.А.. VBA в OFFICE 2000.Офисное программирование. -М.: Издательско-торговый дом "Русская редакция", 1999. - 480 с.
6. Васильев А., Андреев А. VBA в OFFICE 2000. -Спб.: "Питер", 2001. - 432 с.



СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы и успешно реализовал инновационную образовательную программу «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий», что позволило выйти на качественно новый уровень подготовки выпускников и удовлетворять возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях науки. Реализация этой программы создала основу формирования программы дальнейшего развития вуза до 2015 года, включая внедрение современной модели образования.

---

#### **КАФЕДРА ПРИКЛАДНОЙ ЭКОНОМИКИ И МАРКЕТИНГА**

Кафедра прикладной экономики и маркетинга была основана 25 мая 1995 года в связи с началом подготовки в СПбГУ ИТМО бакалавров по направлению 521600 «Экономика». В 1997 году кафедра стала готовить сначала бакалавров, а затем и специалистов по специальности 071900 «Информационные системы в экономике». Со дня основания и по настоящее время кафедрой руководит Почётный работник высшего профессионального образования, доктор экономических наук, профессор, действительный член Российской академии естествознания Олег Валентинович Васюхин.

В настоящее время кафедра прикладной экономики и маркетинга обучает студентов по специальности 080801 «Прикладная информатика в экономике», а также готовит бакалавров и магистров по направлению 080100 «Экономика»

Кадровый состав кафедры представлен специалистами высшей квалификации – три доктора экономических наук, профессора; 9 кандидатов наук, доцентов и 6 старших преподавателей и ассистентов. Около 30% преподавателей – это молодые специалисты, обучающиеся в аспирантуре или недавно закончившие её.

В соответствии с утверждёнными учебными планами, преподаватели кафедры читают более 40 экономико-управленческих и информационных дисциплин как для студентов своих специальностей и направлений, так и для студентов всего университета. С целью обеспечения более эффективного учебного процесса преподавателями кафедры разработаны более 25

учебно-методических пособий, в том числе, часть из них в виде электронных учебников.

Кафедра обладает современной материально-технической базой. Большая часть учебного процесса реализуется в компьютерных классах Гуманитарного факультета, подключённых к сети Интернет. Все виды занятий, текущий контроль знаний, а также разнообразные виды самоподготовки студентов осуществляются на основе балльно-рейтинговой системы организации учебного процесса.

Ежегодно кафедра выпускает около 50 специалистов, бакалавров и магистров, которые успешно работают на предприятиях различных форм собственности и направлений деятельности. Часть выпускников каждый год продолжают обучение в аспирантуре СПбГУИТМО. Практически все студенты кафедры, начиная с 3-4 курса, в свободное время работают на предприятиях Санкт-Петербурга, что в большинстве случаев является основой для прохождения различного рода практик и подготовки выпускной квалификационной работы.

Преподаватели и аспиранты кафедры ведут активную научно-исследовательскую деятельность, участвуя в хоздоговорных исследованиях для предприятий и организаций, а также в крупных госбюджетных НИР. В учебном процессе кафедры принимают участие представители промышленности и науки Санкт-Петербурга.

В настоящее время кафедра прикладной экономики и маркетинга является одной из ведущих выпускающих кафедр Гуманитарного факультета СПбГУ ИТМО.

Петр Васильевич Бураков  
Вадим Юрьевич Петров

## ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ЭКОНОМИКЕ

Учебное пособие

В авторской редакции

Дизайн

В.Ю.Петров

Верстка

В.Ю.Петров

Редакционно-издательский отдел Санкт-Петербургского государственного  
университета информационных технологий, механики и оптики

Зав. РИО

Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99

Подписано к печати \_\_\_\_\_

Заказ № \_\_\_\_\_

Тираж 50

Отпечатано на ризографе

**Редакционно-издательский отдел**  
Санкт-Петербургского государственного  
университета информационных техноло-  
гий, механики и оптики  
197101, Санкт-Петербург, Кронверкский пр., 49

