**Лабораторная работа №13**

**Цель работы:**

.

1. Программирование с использованием функций Excel.
2. Форматирование рабочего листа.
3. Использование комментариев.

*Продолжительность работы – 4 академических часа*

**1. Использование функций Excel.**

Чтобы применить в программе VBA функцию Excel, включите ее в формулу, а затем вставьте в нужную ячейку.

*Финансовые функции*

Встроенные функции Excel разбиты по категориям, перечень которых достаточно обширен. Способ использования функций рассмотрим на примере финансовых функций Excel, предназначенных для проведения различных вычислений, таких как амортизацион-ные отчисления, инвестиционные расчеты и т.п.

Часто используется функция **ПЛТ** (PMT- англоязычный аналог), для вычисления размера выплат по займу.

Например, если вы рассматриваете кредит на покупку автомобиля стоимостью 20000000 в под 5% годовых на четыре года, то эта функция вычислит сумму ежемесячных выплат по такому кредиту.

Синтаксис функции ПЛТ следующий:

ПЛТ(*ставка, кпер, пс, бс, тип)*

• ***ставка***— процентная ставка за период;

• ***кпер***— количество периодов или срок кредита;

• ***пс***— сумма кредита (текущая стоимость);

*•* ***бс***— необязательный аргумент; сумма, которую необходимо выплатить после всех периодических выплат (будущая стоимость). Для большинства кредитов эта сумма равна нулю, и потому нулевое значение используется для данного аргумента по умолчанию;

• ***тип***— необязательный аргумент; равен нулю (по умолчанию), если выплаты осуществляются в конце каждого периода. Равен числу 1, если выплаты осуществляются в начале каждого периода.

Возвращаемое функцией значение — это сумма, которую необходимо выплачивать в каждом периоде. Важно, чтобы для аргументов ***ставка***и ***кпер***использовался один и тот же период. Для большинства кредитов применяется годовое значение процентной ставки, однако выплаты осуществляются ежемесячно. Чтобы решить предложенную выше задачу, необходимо выполнить такие вычисления:

• 5% годовых — это то же самое, что о. 05/12 в месяц;

• период в 4 года — это то же самое, что 4\*12 месяцев.

Формула для вычисления ежемесячных выплат по данному кредиту выглядит так: =ПЛТ(0,05/12, 4\*12, 20000000)

Функция ПЛТ возвращает отрицательное значение для выплат по кредиту, поскольку это денежные затраты, а не денежные доходы. Чтобы получить положительное значение, просто поставьте знак "минус" перед формулой.

Программа, код которой ниже, является примером использования данной функции для создания рабочего листа, на котором вычисляются суммы выплат по кредиту. Кроме того, в этой программе продемонстрирован способ создания пользовательских приложений с помощью возможностей языка VBA. Этот способ состоит в написании программы, создающей рабочий лист вместе со всеми необходимыми заголовками, ячейками для ввода данных и формулами. Будучи выполненной один раз, программа больше не используется, а полученный в результате рабочий лист сам представляет собой пользовательское приложение.

*Программа, создающая рабочий лист для вычисления выплат по кредиту*

Public Sub Кредит()

Dim wb As Workbook

Dim ws As Worksheet

' Создание новой пустой рабочей книги

Set wb = Workbooks.Add

‘ Изменение названия первого рабочего листа

wb.Worksheets(1).Name = "Выплаты по кредиту"

Set ws = wb.Worksheets("Выплаты по кредиту")

‘ Добавление заголовка на рабочем листе

ws.Range("A1") = "Выплаты по кредиту"

' Размещение заголовков в столбце В

ws.Range("B4") = "Сумма кредита"

ws.Range("B5") = "Годовая процентная ставка"

ws.Range("B6") = "Срок кредита в годах"

ws.Range("B7") = "Ежемесячные выплаты"

’ Увеличение ширины столбца В с целью полного

‘ отображения текста заголовков

ws.Range("B1:C1").EntireColumn.AutoFit

‘ Помещение формулы для вычисления выплат по кредиту в ячейку С7

***Формула для расчета***

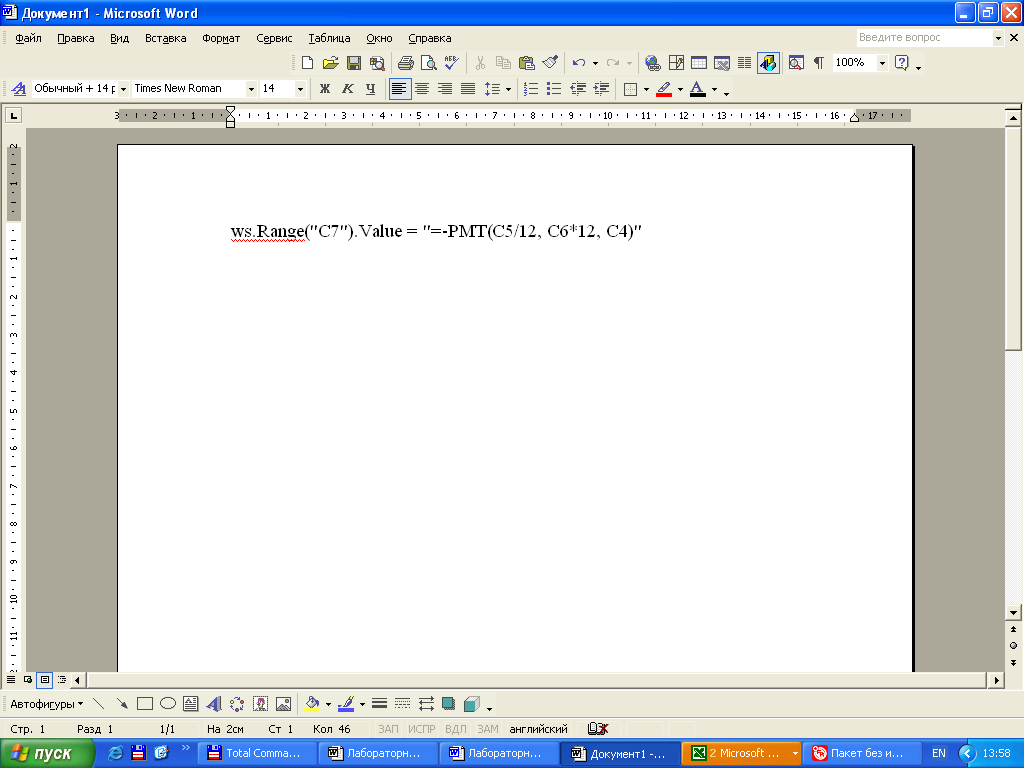
' Сохранение рабочей книги

wb.SaveAs FileName:="Кредитный\_калькулятор"

End Sub

**Чтобы использовать данную программу выполните**:

1. Создайте новую рабочую книгу и сохраните ее в рабочей папке с именем **LAB\_5**
2. Откройте редактор Visual Basic и добавьте в проект модуль и переименуйте его, дав имя КРЕДИТ.
3. Копируйте текст процедуры в тексте лаб. работы. Откройте окно редактирования кодов модуля и вставьте текст процедуры.
4. Вместо ***Формула для расчета*** наберите текст приведенной ниже формулы



1. Сохраните файл, закройте окно редактора.
2. Запустите программу одним из известных вам способов.
3. Проверьте работоспособность, вводя реальные значения параметров функции **ПЛТ**.
4. Результаты предъявите преподавателю.

Способ отображения результатов работы данной программы достаточно прост и требует внесения изменений:

- применение форматирования может сделать результаты более наглядными и привлекательными;

- сумма кредита должна отображаться в денежном формате, а ставка в процентном;

- по неосторожности можно стереть или изменить формулу, вычисляющую результат.

Далее мы еще вернемся к этой программе и устраним очевидные недостатки.

**2. Форматирование рабочего листа**

**2.1. Основные понятия. Описание свойств и методов, используемых при форматировании.**

*Рассматриваемый в этом разделе материал далее будет использован при создании программ различного назначения. Это обстоятельство потребует с вашей стороны затрат времени и усилий на усвоение несложных операторов. Не переходите к разделу выполнить…, в любом случае вы вернетесь к этому материалу содержащему описание применяемых объектов, свойств и методов.*

Термин ***форматирование***относится к внешнему виду рабочего листа. Поскольку форматирование рабочего листа никак не влияет на особенности его функционирования, некоторые считают саму возможность форматирования несущественным дополнением, которому не стоит уделять особого внимания.

Это большая ошибка! Важным аспектом любого рабочего листа является его простота и дружелюбность по отношению к пользователю. Благодаря удачному применению форматирования рабочий лист не только становится более удобным в использовании, но при этом уменьшается также вероятность возникновения ошибок. В настоящем уроке рассматриваются наиболее важные аспекты форматирования рабочих листов.

**Форматирование шрифтов**

В VBA для форматирования шрифтов используется объект **Font**, имеющий ряд свойств, соответствующих различным параметрам шрифтов, основные из них описаны в таблице

Таблица 5.1.

|  |  |
| --- | --- |
| *Свойство* | *Описание* |
| *Name* | Название шрифта |
| *Color* | Цвет шрифта , указываемый в значениях RGB |
| *Size* | Размер шрифта в пунктах (1 пункт равен 1/72 дюйма |
| *Bold* | Полужирное начертание |
| *Italic* | Курсив |

Свойство *Color* использует значение RGB, которое определяет требуемый цвет путем фиксирования его красной (Red), зеленой (Green) и синей (Blue) составляющих.

Чтобы присвоить свойству *Color* значение используется функция RGB(r,g,b). Значение каждого аргумента может меняться от 0 до 255 и передать более 16 миллионов оттенков от RGB(0,0,0) – черный цвет

до RGB(255,255, 255) – белый цвет

В зависимости от значений аргументов, каждый из которых может меняться от 0 до 255, можно воспроизвести более 16 миллионов оттенков.

Очевидные цвета это:

RGB(255,0, 0) – красный

RGB(0,255, 0) – зеленый

RGB(0,0, 255) – синий

Например, показанная ниже строка кода делает шрифт выделенных ячеек желто-зеленым

**Selection.Font.Color = RGB(120,255, 0)**

В языке VBA предусмотрен ряд стандартных констант для представления некоторых цветов. Это vbBlack, vbRed, vbGreen, vbYellow, vbBlue, vbMagenta, vbCyan и vbWhite.

Эти константы можно использовать вместо функции RGB для изменения свойств Color.

Например **- Selection.Font.Color =** **vbYellow.**

**Выравнивание содержимого ячеек**

Чтобы изменить способ выравнивания, используйте следующие свойства объекта Range.

• HorizontalAlignment. Принимает значения xlLeft (горизонтальное выравнивание по левому краю), xlCenter (по центру) или xlRight (по правому краю).

• VerticalAlignment. Принимает значения xlTop (вертикальное выравнивание по верхнему краю), xlCenter (по центру) или xlBottom (по нижнему краю).

Например, показанная ниже процедура выравнивает содержимое выделенных ячеек

по горизонтали и вертикали

Public Sub Centr()

Dim r As Range

Set r = Selection

With Selection

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

End With

End Sub

**Границы ячеек**

Excel может добавлять обрамление вокруг отдельных ячеек или их группы. Для диапазонов из двух и более ячеек такое обрамление может содержать как рамку по внешнему краю, так и границы между ячейками внутри диапазона. Каждый объект Range имеет коллекцию Borders, значения которой определяют характер используемых для данного диапазона границ. Каждая отдельная граница представлена своим объектом **Border**, ей соответствует определенная константа (табл. 5.2).

***Таблица 5.2 . Границы диапазона и соответствующие им константы***

|  |  |
| --- | --- |
| ***Граница*** | ***Константа*** |
| Нижняя граница диапазона | xlEdgeBottom |
| Верхняя граница диапазона | xlEdgeTop |
| Левая граница диапазона | xlEdgeLeft |
| Правая граница диапазона | xlEdgeRight |
| Вертикальные границы внутри диапазона | xllnsideVertical |
| Горизонтальные границы внутри диапазона | xlInsideHorizontal |

Работать с границами диапазона можно двумя способами.

- Если используется коллекция **Borders** без аргументов, изменению подвергаются все границы диапазона.

- Применяя коллекцию **Borders** с одной из констант, представленных в табл. 5.2, в качестве аргумента, вы воздействуете только на соответствующую границу.

Например, следующая строка добавляет двойную границу по всем четырем краям текущей выделенной области и внутри нее:

Selection.Borders.LineStyle = xlDouble

Напротив, эта строка кода добавляет двойную границу только по верхнему краю выделенной области:

Selection.Borders(xlEdgeTop).LineStyle = xlDouble

Параметры отображения границ, которые можно изменять в программе, приведены в табл. 5.2

***Таблица 5.3. Свойства объекта Border***

|  |  |  |
| --- | --- | --- |
| ***Свойство*** | ***Описание*** | ***Значения*** |
| Color | Цвет границы | Значение RGB (см. ранее); по умолчанию используется черный цвет |
| LineStyle | Стиль границы, например: непрерывная линия,пунктирная линия, линия из точек и т.д. | xlContinuous, xlDash, xlDashDot, xlDashDotDash, xlDot, xlDouble, xlSlantDashDot |
| Weight | Толщина границы: от очень тонкой до очень широкой | xlHairLine, xlThin, xlMedium, xlThick |

Чтобы удалить границу, присвойте ее свойству LineStyle значение xlLineStyleNone.

**Установка фона ячеек**

Фон ячейки или диапазона устанавливается изменением свойств объекта **Interior.** Свойству этого объекта **Color** присваивается значение RGB(рассмотрено ранее).

После ознакомления с инструментами форматирования, применим их для улучшения внешнего вида представляемых на рабочем листе данных

**2.2. Практические задания и порядок их выполнения**

**Задание 1**

**Создать таблицу представленную на рис. 1, используя форматирование преобразовать ее к виду, показанному на рис.2**

Рассмотрим три различных варианта выполнения задания.

***1 вариант*** **-** преобразуем таблицу, используя команды и панель инструментов Excel.

***2 вариант -*** создадим макрос выполняющий требуемое форматирование

***3 вариант -*** разработаем программу с использованием приемов форматирования, рассмотренных ранее в этой работе.

**Предварительно выполните:**

1. Откройте файл LAB\_5.
2. На листе 1 создайте таблицу (данные представленные на Рис. 1).
3. Копируйте таблицу и на листе 2 разместите три копии, пример приведен на рис.3. Переименуйте лист 2, новое имя **Формат.**
4. Сохраните файл.

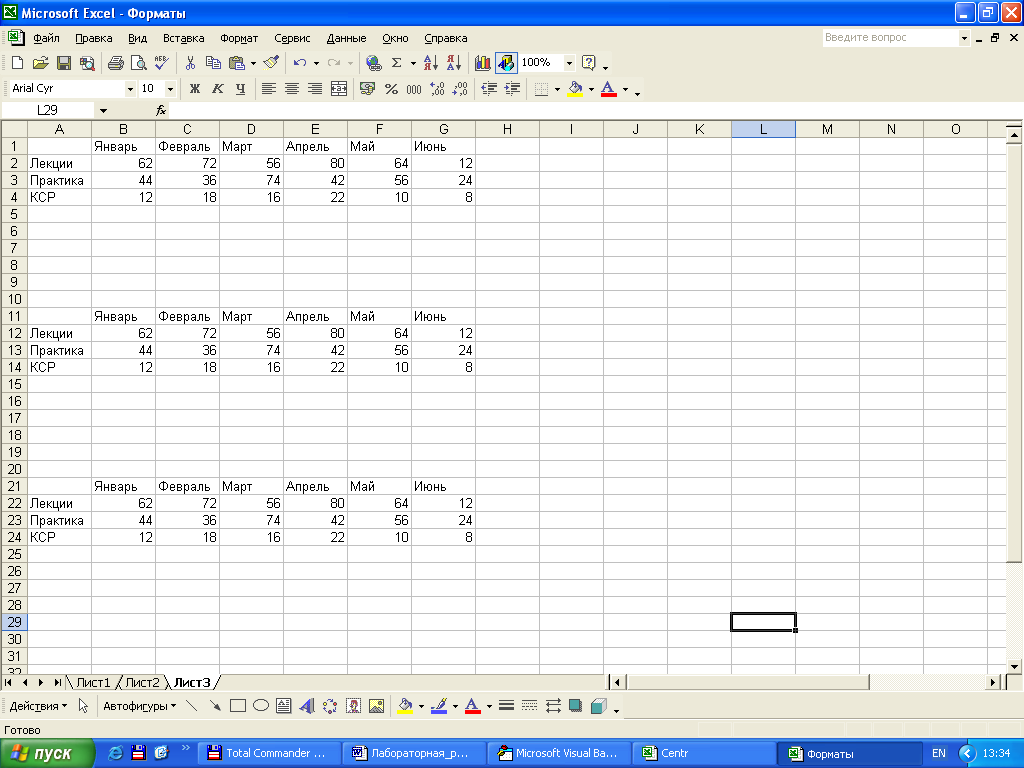
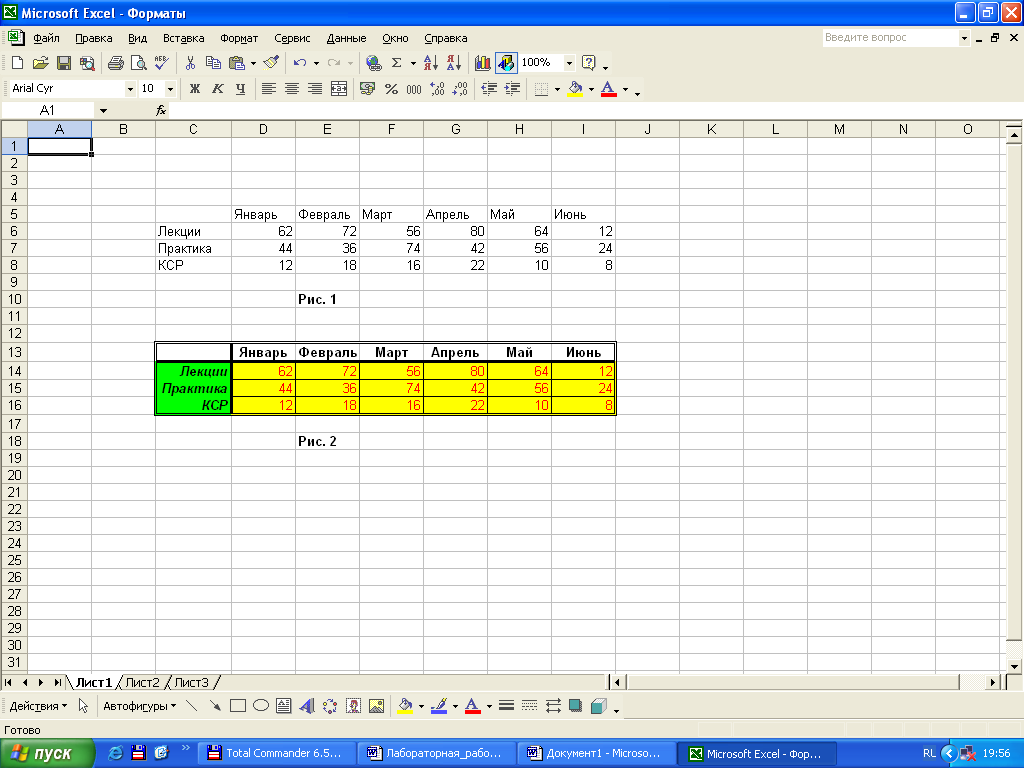


Рис.3

Далее рассмотрим три варианта выполнения форматирования таблицы в каждом из них необходимо установить параметры (см. рис.2):

- толстая линия границы под верхней строкой

- толстая линия границы справа от первого столбца

- тонкие линии границ между ячейками в области размещения числовых данных

- двойная линия границы по периметру диапазона размещения таблицы

- текст в верхней строке выделен полужирным шрифтом и выровнен по центру

- текст в левом столбце выделен полужирным шрифтом и выровнен по правому краю.

- цвет шрифта в ячейках с цифровыми данными – *красный*

- фоновый цвет ячеек с названиями строк в первом столбце - ***зеленый***

- фоновый цвет ячеек с цифровыми данными - ***желтый***

**Вариант 1**

Форматирование с использованием команд и кнопок на панели инструментов.

**Выполните:**

1. На **листе 2** выберите одну из нижних копий.
2. Используя показанные на рис.2 параметры и их описание форматировать выбранную копию таблицы.
3. После завершения форматирования файл сохранить, результаты представить преподавателю.

**Вариант 2**

Создание макроса выполняющего требуемое форматирование

**Выполните:**

1. На **листе 2** выберите верхнюю копию.

2. Перейти в режим записи макроса, макросу присвоить имя Form\_1.

3. Выполнить форматирование.

4. После завершения форматирования остановить запись макроса, файл сохранить.

5. Проверить работоспособность макроса, предварительно на место верхней копии таблицы, которая использовалась для создания макроса вставить исходный вариант без элементов форматирования.

6. Результаты представить преподавателю

При записи макроса с учетом опыта выполнения варианта 1, попытайтесь свести к минимуму ошибочные действия. Каждый шаг фиксируется и отображается в кодах VBA.

Наряду с преимуществами, которые появляются при наличии макроса, ускоряется выполнение пакета команд, имеются недостатки. Разработанный нами макрос форматирует таблицу, которая имеет фиксированные размеры и положение на листе.

Далее мы выполним редактирование текста макроса, чтобы снять ограничение по размещению таблицы на рабочем листе.

**Вариант 3**

Создание программы выполняющей требуемое форматирование.

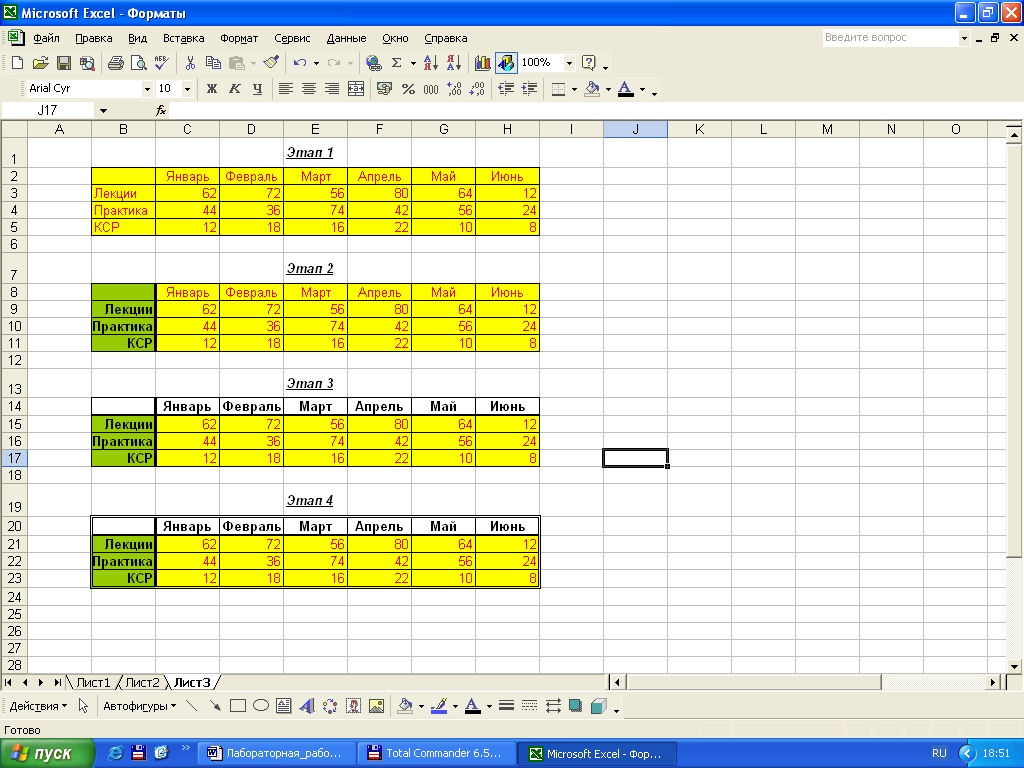
На рис. 4 графически отбражены этапы преобразования таблицы, которые должны быть реализованы в программе с помощью операторов VBA

***Этап 1*** – выделить диапазон, установить желтый фон, шрифт – красный, границы ячеек – тонкие линии

***Этап 2*** – выделить первый столбец, установить зеленый фон, шрифт – полужирный черный, толстая правая граница.

***Этап 3***– выделить первую строку, установить белый фон, шрифт – полужирный черный, толстая нижняя граница.

***Этап 4***– выделить диапазон, установить обрамление внешних границ двойной линией



**Рис. 4**

Далее представлен текст программы, включающий основные структурные элементы.

Тексты помеченные *подчеркнутым курсивом* после копирования и использования в работе следует удалить или заменить операторами.

Public Sub FormatTable()

' Форматирует таблицу, в которой находится активная ячейка

Dim r As Range, r1 As Range

' Проверка наличия диапазона

If TypeName(Selection) <> "Range" Then

MsgBox "Чтобы программа работала сделайте активной любую ячейку диапазона"

Exit Sub

End If

' Выделение области, содержащей всю таблицу*(Этап 1 рис.4)*

Set r = Selection.CurrentRegion

' Добавление тонких границ ячеек во всем диапазоне

***1***. r.Borders.LineStyle = xlContinuous

***2***. r.Borders(xlInsideHorizontal).Weight = xlThin

' Выбор желтого фона, цвет шрифта красный

***3.*** r.Interior.Color = vbYellow

***4.*** r.Font.Color = vbRed

' Выделение первого столбца диапазона *(Этап 2)*

Set r1 = r.Columns(1)

' Добавление толстой правой границы столбца

*Вставьте два оператора подобные* ***1 и 2*** *внесите изменения в значения*

*аргументов, а диапазон* ***r*** *замените на* ***r1***

' Выбор зеленого фона, цвет шрифта черный

*Операторы подобные* ***3*** *и* ***4*** *замените* ***r*** *на* ***r1*** *и константы для представления цвета*

' Выделение текста жирным шрифтом выравнивание справа

***5***. r1.Font.Bold = True

***6.*** r1.HorizontalAlignment = xlRight

' Выделение первой строки диапазона *(Этап 3)*

Set r1 = r.Rows(1)

' Добавление толстой нижней границы строки.

r1.Borders(xlEdgeBottom).LineStyle = xlContinuous

r1.Borders(xlEdgeBottom).Weight = xlThick

' Выбор белого фона, цвет шрифта черный

*Используйте операторы Этапа 2, замените цвет фона*

' Выделение текста жирным шрифтом выравнивание по центру

*Операторы подобные* ***5*** *и* ***6***

' Выделение всего диапазона и обрамление двойной линией по внешним границам*(Этап 4)*

Set r = Selection.CurrentRegion

r.Borders(xlEdgeTop).LineStyle = xlDouble

*Добавьте еще три таких же оператора, в которых отличается константы соответствующие*

*верхней, левой и правой границам (см. таблица 5.2)*

End Sub

**Выполните:**

1. Откройте файл **LAB\_5**
2. Откройте редактор Visual Basic. Модуль в котором сохранен созданный ранее макрос переименуйте, дав имя **ФОРМАТ**.
3. Копируйте текст процедуры **FormatTable** в тексте лаб. работы.
4. Откройте окно редактирования кодов модуля и вставьте текст процедуры.
5. Редактируйте текст процедуры, удаляя или заменяя операторами выделенные подчеркнутым курсивом фрагменты текста.
6. Сохраните файл **LAB\_5.**
7. Закройте окно редактора и откройте лист **Формат.**
8. Сделайте активной любую ячейку в копии исходной таблицы и запустите процедуру **FormatTable.**
9. Проверьте возможность использования созданной программы для форматирования таблиц другого размера**.** ( Сделайте копию исходной таблицы и добавьте строки и/или столбцы).
10. Результаты представьте преподавателю

**Задание 2**

Редактировать макрос **Form\_1,** чтобы обеспечить возможность форматирования таблицы расположенной в любом месте рабочего листа.

Напомним, что макрос позволяет форматировать таблицу, расположенную в левом верхнем углу рабочего листа. Причем таблица может быть пустая (можете проверить).

Измените текст процедуры реализующей макрос таким образом, чтобы форматирование проводилось в предположении, что левый верхний угол редактируемой области совпадает с активной ячейкой.

**Выполните:**

1. Откройте текст макроса **Form\_1** для редактирования. (Можно использовать окно **Макрос** с выбором вкладки **Изменить** или использовать окно редактора **VBA).**
2. Вставьте две строки чтобы они предшествовали программным кодам

**Dim r1 As Range**

**Set r1 = ActiveCell**

1. В тексте макроса все ссылки на диапазоны с использованием объекта **Range** изменить подобно следующему примеру**:**

**Range("D5:I7").Select** на **r1.Range("D5:I7").Select.** (добавляется **r1**).

*Эти изменения выделяют диапазон не от левого верхнего угла рабочего листа, а от активной ячейки.*

**4.** Сохранить файл, перейти на любой лист **Excel** и проверить работу макроса.

**Задание 3**

После изучения средств форматирования, сделаем более наглядной таблицу расчета выплат по кредиту. Внесем в текст процедуры Кредит изменения для выполнения следующих действий:

- заголовок рабочего листа в ячейке А1 отображаются более крупным шрифтом;

- тексты в ячйках В4:В7 выделить полужирным шрифтом;

- для ячеек С4:С7 установить соответствующий числовой формат;

- вся область таблицы должна иметь серый фон, а ячейки для ввода данных- белый.

**Выполнить:**

1. Открыть для редактирования текст программы **Кредит.**
2. Внесите в текст программы изменения, которые представлены в ниже представленном листинге. Новый код, добавляемый в исходный текст, обрамлен комментариями ‘\*\*\*.
3. Сохраните файл, закройте окно редактора и запустите программу **Кредит.**
4. Результаты предъявите преподавателю**.**

*Выплаты по кредиту и форматирование*

Public Sub Кредит ()

Dim wb As Workbook

Dim ws As Worksheet

' Создание новой пустой рабочей книги

Set wb = Workbooks.Add

' Переименование первого рабочего листа.

wb.Worksheets(1).Name = "Выплаты по кредиту"

Set ws = wb.Worksheets("Выплаты по кредиту")

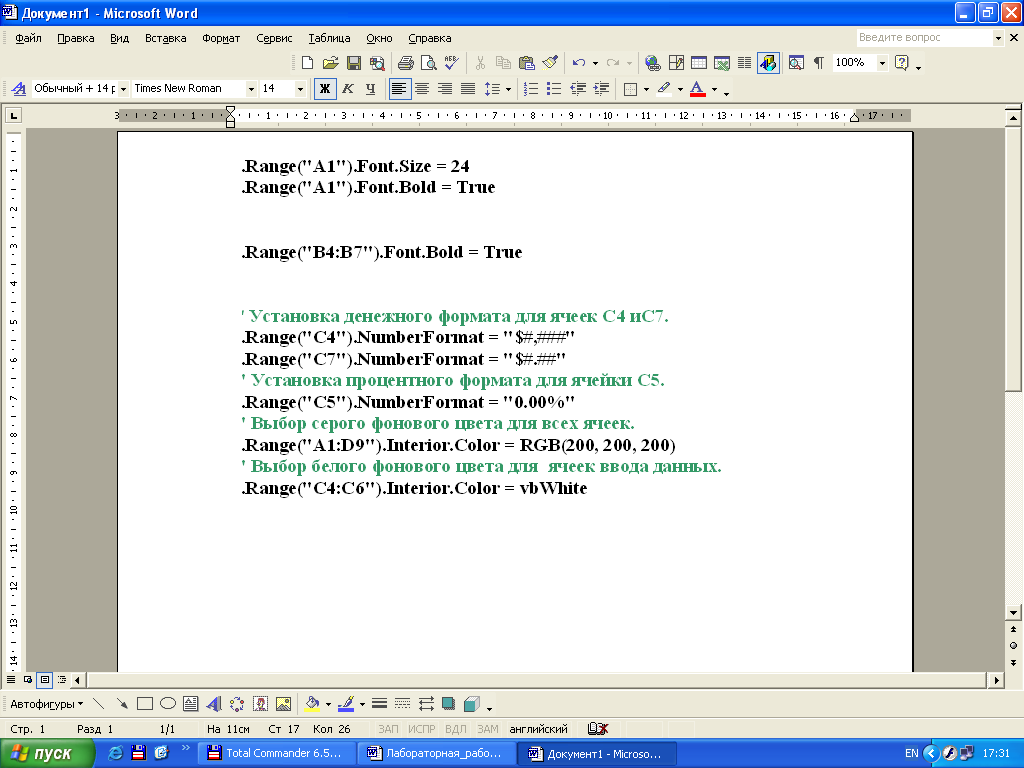
With ws

' Добавление заголовка на рабочем листе.

.Range("A1") = "Выплаты по кредиту"

'\*\*\*

**' Изменение шрифта ячейки А1**



'\*\*\*

' Размещение заголовков в столбце В.

.Range("B4") = "Сумма кредита"

.Range("B5") = "Годовая процентная ставка"

.Range("B6") = "Срок кредита в годах"

.Range("B7") = "Ежемесячные выплаты"

'\*\*\*

**'Изменение шрифта в B4:B7 на жирный.**

**.Range("B4:B7").Font.Bold = True**

'\*\*\*

' Установка автоматического выбора ширины столбца В

' для полного отображения текста заголовков.

.Range("B1").EntireColumn.AutoFit

' Помещение формулы для выплат по кредиту в ячейку C7

.Range("C7").Value = "=-PMT(C5/12, C6\*12, C4)"

'\*\*\*

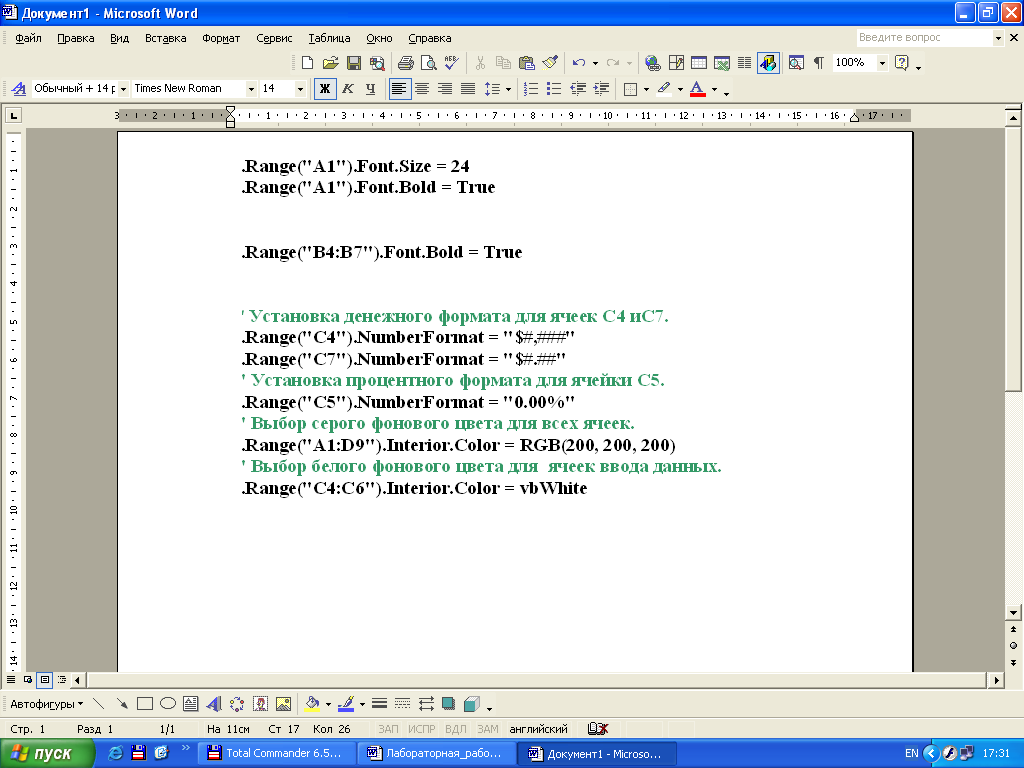
' **Установка денежного формата для ячеек С4 иС7.**

**.Range("C4").NumberFormat = "$#,###"**

**.Range("C7").NumberFormat = "$#.##"**

**' Установка процентного формата для ячейки С5.**

**.Range("C5").NumberFormat = "0.00%"**



'\*\*\*

End With

' Сохранение рабочей книги.

wb.SaveAs Filename:="Кредитный калькулятор"

End Sub

**Задания для самостоятельной работы.**

**При выполнении заданий используются навыки, инструменты и операторы, освоенные при выполнении лабораторных работ 2, 3, 4 и 5.**

Разрабатываются процедуры, позволяющие создавать простую базу данных. При разработке программ, необходимо предусмотреть выполнение следующих операций:

* Построение заголовка таблицы с помощью макроса
* Заполнение строк таблицы (записей БД)
* Форматирование таблицы

Вид таблицы, для создания которой предназначено разрабатываемое приложение, представлен на рис. 1

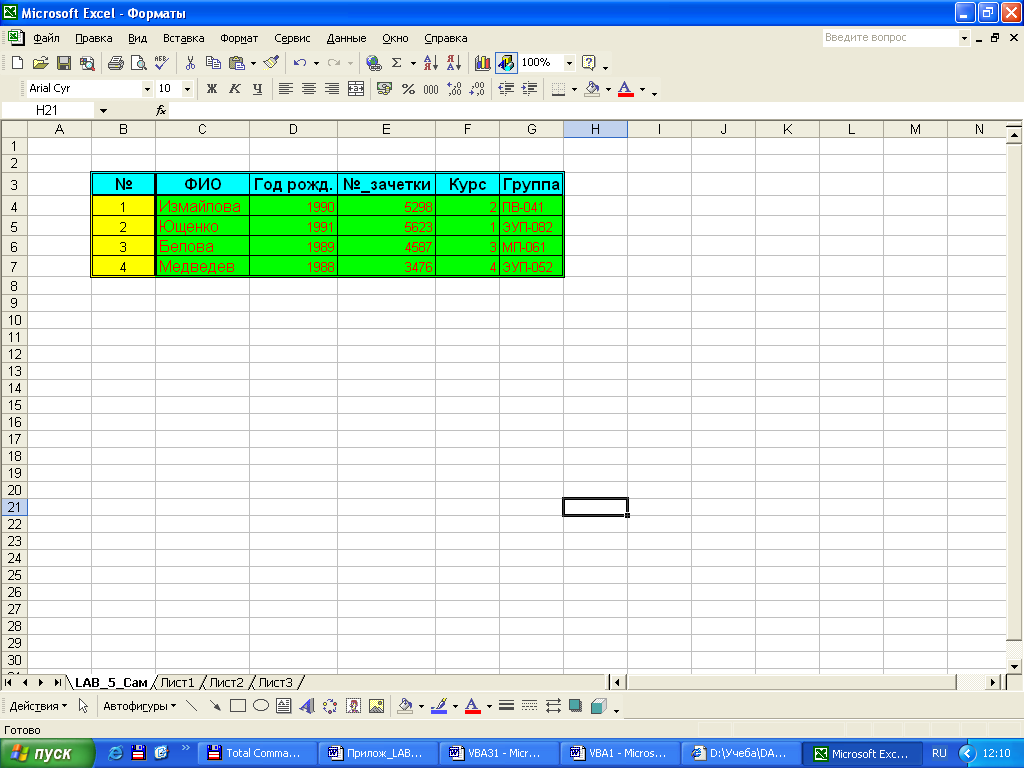


Рис. 1

**Выполните:**

**Откройте Excel, файл для выполнения результатов данной работы сохраните в рабочей папке с именем MY\_WORK**

**Задание 1. Создайте макрос для построения заголовка таблицы.**



- Макрос сохранить с именем **Титул.**

- Ввод заголовка начинается с ячейки **А1.**

**-** После завершения набора строки сделайте активной ячейку **А2** и остановите запись макроса, файл сохраните.

- Редактируйте текст макроса, чтобы заголовок строился на рабочем листе, начиная с активной ячейки.

**-** Проверьте работу макроса.

**Задание 2 Создайте процедуру для ввода строк таблицы**

- Алгоритм подобной процедуры рассмотрен в лабораторной работе №4 (см. программа RR\_2).

- Разрабатываемая процедура будет запускаться после запуска макроса **Титул.**

- Процедуру разместите в модуле, содержащем макрос **Титул.** Имя создаваемой процедуры **Таблица.**

- После создания процедуры, проверьте ее работоспособность, введите 4..5 строк.

**Задание 3. Создайте процедуру для форматирования таблицы.**

Форматирование выполните в соответствии с приведенным образцом на рис.1

Первая строка: шрифт полужирный, расположение текста - по центру, нижняя граница –толстая линия красного цвета.

Первый столбец- шрифт полужирный синего цвета, расположение текста – слева, правая граница - двойная линия, цвет фона - желтый.

Таблица обрамлена двойной линией, границы ячеек внутри таблицы – тонкие линии

Процедуру разместите в модуле, содержащем макрос **Титул.** Имя создаваемой процедуры **Формат\_Т.**

Проверьте работу процедуры, форматируйте ранее созданную таблицу.

**Задание 4**

**Задание 4. Создайте процедуру для форматирования диапазона таблицы, за исключением первой строки и первого столбца**:

Цвет фона – зеленый, шрифт - красный

Процедуру разместите в модуле, содержащем макрос **Титул.** Имя создаваемой процедуры **Формат\_Д.**

Пример подобной процедуры см. пункт **3** в разделе ***Дополнительная информация***

Проверьте работу созданных процедур, для чего перейдите на лист не содержащий записей и запустите поочередно Титул⇒Таблица ⇒ Формат\_Т.

***Дополнительная информация для выполнения самостоятельной работы***

**1. Ввод информации в диалоговом режиме**.

Public Sub PR\_2()

Dim rep As Integer

rep = 0

j = ActiveCell.Row - 1

Do While rep <> 7

With Selection

i = ActiveCell.Row

.Offset(0, 0).Value = i – j

.Offset(0, 1).Value = InputBox("Ввод ФИО”)

.Offset(0, 2).Value = InputBox("")

.Offset(0, 3).Value = InputBox("")

.Offset(0, 4).Value = InputBox(".")

.Offset(0, 5).Value = InputBox(".")

.Offset(1, 0).Select

End With

rep = MsgBox("Продолжить ввод", vbYesNo, " Внимание ")

Loop

End Sub

**2. Получение ссылки на таблицу, в которой находится активная ячейка. Форматирование таблицы.**

При выполнении работ связанных с форматированием таблицы используются свойства и методы:

1. Свойство **CurrentRegion** объекта **Range.**

**-** Возвращает ссылку на диапазон, ограниченный пустыми строками, а также столбцами и границами рабочего листа.

1. Свойство **Rows(n)** объекта **Range.**

**-** Возвращает ссылку на n-ю строку диапазона

1. Свойство **Columns(n)** объекта **Range.**

**-** Возвращает ссылку на на n-й столбец диапазона

Public Sub FormatTable\_2()

Dim r As Range, r1 As Range

**Set r = Selection.CurrentRegion**

r.Borders.LineStyle = xlContinuous

r.Borders.Weight = xlThin

r.Interior.Color = vbYellow

r.Font.Color = vbRed

Set r1 = r.Columns(1)

r1.Borders(xlEdgeRight).LineStyle = xlContinuous

r1.Borders(xlEdgeRight).Weight = xlThick

r1.Interior.Color = RGB(0,250,0)

r1.Font.Color = RGB(0,0,0)

r1.Font.Bold = True

r1.HorizontalAlignment = xlRight

Set r1 = r.Rows(1)

r1.Borders(xlEdgeBottom).LineStyle = xlContinuous

r1.Borders(xlEdgeBottom).Weight = xlThick

r1.Interior.Color = vbWhite

r1.Font.Color = vbBlack

r1.Font.Bold = True

r1.HorizontalAlignment = xlCenter

Set r = Selection.CurrentRegion

r.Borders(xlEdgeTop).LineStyle = xlDouble

r.Borders(xlEdgeBottom).LineStyle = xlDouble

r.Borders(xlEdgeLeft).LineStyle = xlDouble

r.Borders(xlEdgeRight).LineStyle = xlDouble

End Sub

**3. Ссылка на диапазон таблицы, из которого исключены первая строка и первый столбец. Форматирование цвета фона и шрифта диапазона.**

Sub ShowAdr()

Dim r As Range, Table As Range

Dim s As String, s1 As String

Dim kr As Integer, kc As Integer

Set Table = ActiveCell.CurrentRegion

With Table

Rem **kr/kc** – количество строк/столбцов в диапазоне

kr = .Rows.Count

kc = .Columns.Count

Rem **s/s1**– адреса ячеек

s = .Cells(2, 2).Address

s1 = .Cells(kr, kc). Address

End With

Set r = Range(s + ":" + s1)

r.Interior.Color = RGB(150, 150, 150)

r.Font.Color = RGB(0, 0, 200)

MsgBox (s + ":" + s1)

End Sub