

«Национальный открытый институт»

**РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТРАСЛЕВОЙ
НАПРАВЛЕННОСТИ
(интегрированная среда VBA)**

**Методические указания
к выполнению практических занятий**

Рекомендовано Методической комиссией по качеству
Национального открытого института
для студентов, обучающихся по направлению
09.03.03 «Прикладная информатика»

Санкт-Петербург
2016

УДК 519.2.06 (07)
ББК 32.973
С34

Методические указания разработаны на основе рабочей программы “Прикладная информатика (по отраслям)” в соответствии с требованиями государственных образовательных стандартов среднего профессионального образования.

Практические задания выполняются студентами самостоятельно по индивидуальному варианту в интегрированной среде VBA. Методические указания предназначены для студентов специальности – 090205 Прикладная информатика (по отраслям).

УДК 519.2.06 (07)
ББК 32.973

© Сибирев В.Н. 2016
© Рачева Н.В. 2016
© «Национальный открытый институт» 2016
© ИКЦ 2016

1. Теоретические сведения

Программирование в интегрированной среде VBA

Преимущество VisualBasicApplication (VBA) по отношению к другим программным средствам не только в простоте, но и в возможности использования всех готовых офисных приложений с многочисленным набором исполнительных функций [4]. Изучение основ языка VBA начинается на первом курсе в рамках дисциплины информатики, поэтому здесь приведены конкретные примеры для более углубленного знакомства со средой VBA [1,2].

Функции оперативного обмена информацией в среде VBA

Функция **MsgBox**

MsgBox(сообщение[,кнопки][,заголовок][, контекст])

На экран выводится диалоговое окно, содержащее сообщение, устанавливается режим ожидания нажатия кнопки пользователем, возвращается значение типа Integer .

Функция **InputBox**

InputBox(сообщение[, заголовок][, умолчание][, Xпоз][,Yпоз][,файл справки, контекст])

На экран выводится диалоговое окно, содержащее сообщение и поле для ввода информации, устанавливается режим ожидания ввода текста пользователем или нажатие кнопки, возвращается значение типа String.

Пример:

Вычислить сумму значений элементов целочисленного одномерного массива из 10 элементов.

```
Sub pr()
```

```
Dim b(10) As Integer
```

```
Dim s As Integer, i As Integer
```

```
s=0
```

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

```
    b(i) = InputBox("Введите число")
```

```
s = s + Val(b(i))
```

 ‘ **Val** – функция преобразования типа переменной «символа – число»

```
Next  
MsgBox (" Сумма=" & s )  
End Sub
```

Объявление массивов

Массив – совокупность однотипных данных, размещенных под одним

именем в оперативной памяти. Доступ к отдельным элементам массива

выполняется по индексу (индексам, если массив - многомерный).

Индекс элемента указывается в круглых скобках после имени массива.

Например:

Massiv(1), Massiv(2), Massiv(7) являются 1-ым, 2-ым, 7-ым элементами

массива с именем Massiv.

Dim C(1 To 10) As Integer

В этой конструкции задается массив из 10 элементов, но индекс

элементов массива изменяется от 1 до 10.

Dim B(10, 10) As Integer

Данная описательная конструкция указывает на двумерный массив

размерностью 11*11.

Пользовательский интерфейс (IDE – IntegratedDevelopmentEnvironment)

Интерфейс — это внешняя среда программного приложения, выполняющая связь (удобный доступ) к программному продукту, что гарантирует и его коммерческий успех.

Проектирование интерфейса для пользователя — процесс циклический. На этом этапе разработки программного приложения желательно чаще общаться с пользователями (заказчиками программного приложения) для выработки наиболее приемлемых по эффективности, удобству и внешнему виду интерфейсных решений.

В качестве стандарта (образца для "подражания") интерфейсных решений может служить любое из приложений MS — Word, Excel или другие. Под решениями подразумеваются дизайн форм, распределение элементов управления, их взаимное расположение.

Для написания исходного текста программы рекомендуется использовать текстовый редактор (как правило, совместимый с Word) встроенный в среду программирования, в данном случае, VBA. Основой является среды VBA транслятор интерпретирующей системы, его роль - перевод с языка VBA в двоичный код выполнения программы.

Интегрированная среда разработки VBA

Для написания программных кодов в **VisualBasic** используется редактор кода. Чтобы его запустить, необходимо в окне **ProjectExplorer** установить курсор на форму (**Userform**) или на модуль (**Models**), для которого создается код, и выполнить одно из следующих действий:

в меню ~ **View** (Вид) выбрать команду ~ **Code** (Код);

из контекстного меню модуля выбрать команду **ViewCode**.

При выполнении любого из этих действий открывается окно редактирования (рис. 1), в котором можно набирать текст программы. Проектирование связи «пользователь – программный код» через **Userform**.

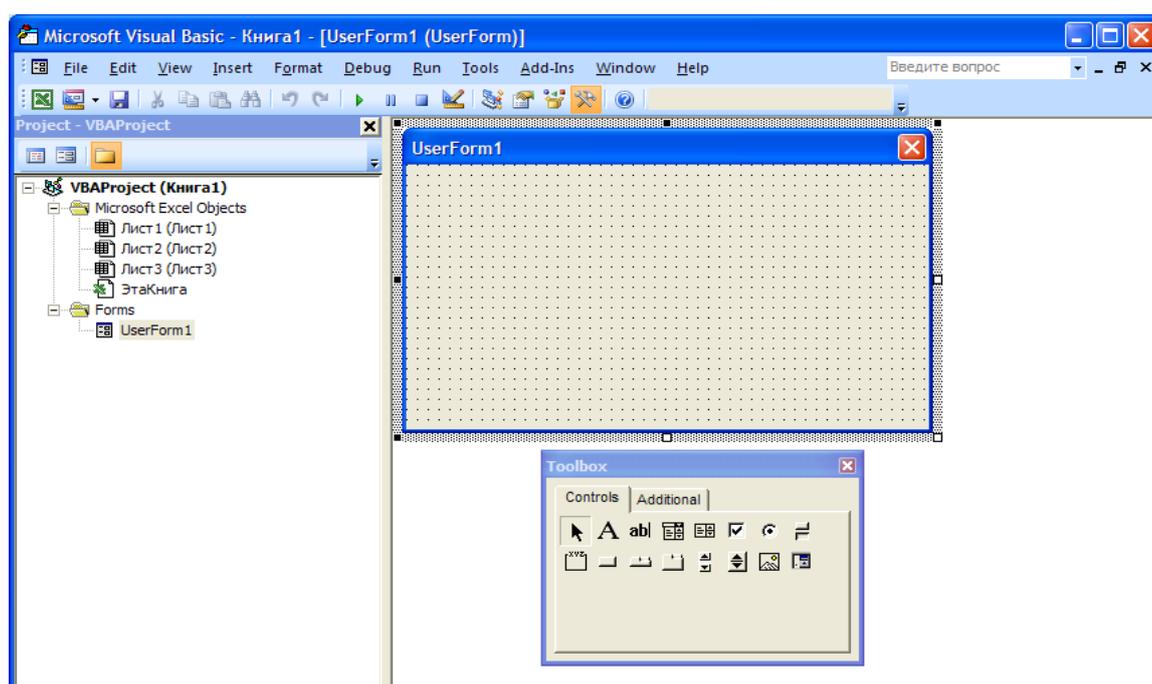


Рис.1. Создание формы с помощью инструментария ToolBox.

Элементы управления ToolBox	Использование/Описание ToolBox
Поле (EditBox)	Отображает данные и используется для ввода информации
Надпись (Label)	Отображает неизменяемый текст в формах, например, заголовки для других элементов управления
Флажок (CheckBox)	Позволяет задать значение "Истина" или "Ложь"
Поле со списком (ComboBox)	Используется либо для выбора нужного элемента из списка, либо для ввода значения в поле
Список (ListBox)	Позволяет создать список, содержащий требуемые значения
Кнопка (CommandButton)	Создает кнопку, при нажатии которой выполняется команда
Переключатель (OptionButton)	Используется для предоставления выбора одного варианта из многих
Рамка (Frame)	Позволяет установить графическую или функциональную группировку элементов управления

Визуальное проектирование пользовательского интерфейса

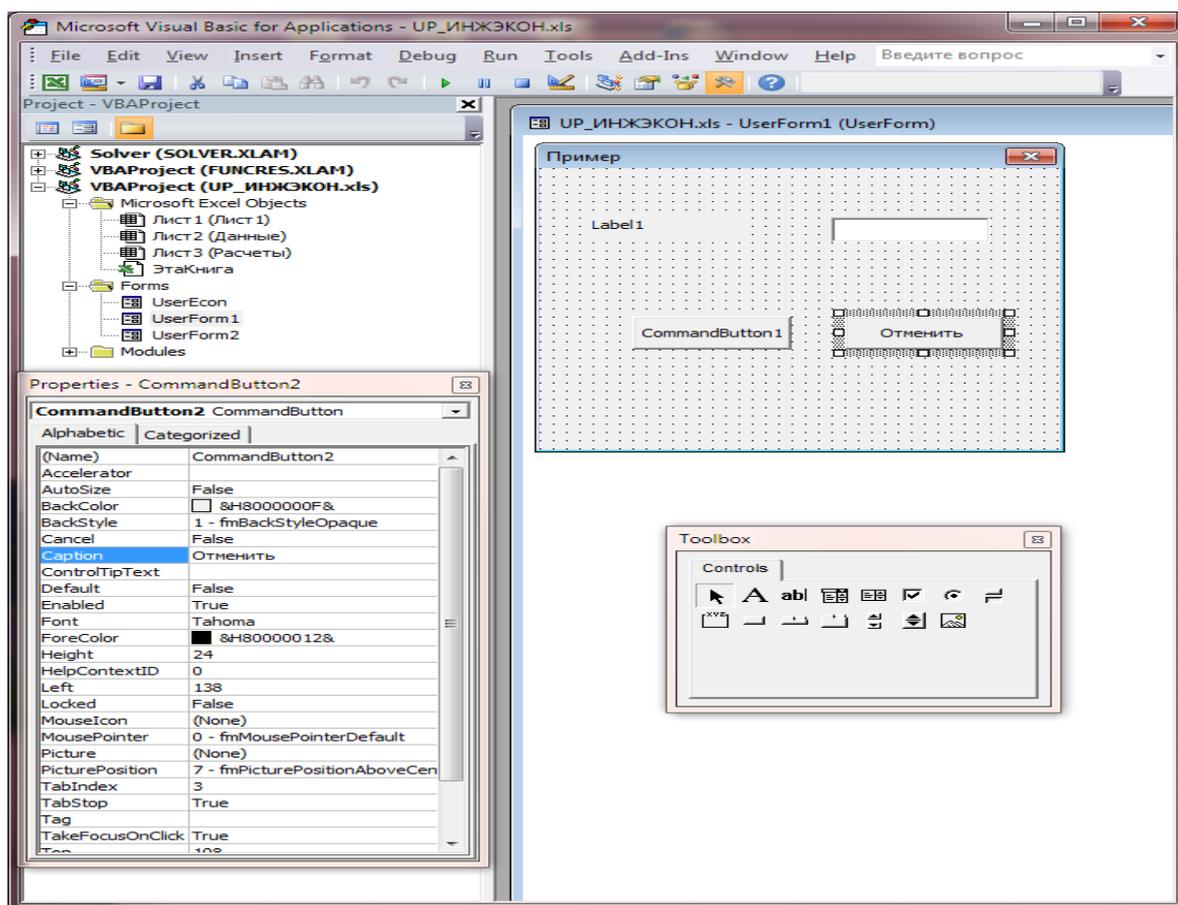


Рис.2. Пример проекта простой формы «пользователь – программный код».

Интерфейс должен быть интуитивно понятным. Желательно, чтобы все действия легко запоминались и не требовали утомительных процедур: выполнения дополнительных команд, лишних нажатий на кнопки, вызова промежуточных диалоговых окон.

Редактор **VBA** включает набор программных средств, которые значительно облегчает процесс кодирования. Эти средства выполняют анализ введенного текста и предлагают только допустимые типы данных, встроенные шаблоны процедур, функций и т.п.

Рассмотрен пример расчета суммы целых чисел. Данные размещены на рабочем листе с именем «Данные». Результат отображен на рабочем листе с именем «Расчеты» (см. рис.5). По команде «Выполнить» происходит запуск приведенного ниже программного кода.

```

Private Sub CommandButton1_Click()
Dim Mas(50), s As Single
Dim LD, LR As Object
Dim i, n As Integer
Set LD = Sheets("Данные")
Set LR = Sheets("Расчеты")
'n – количество элементов массива, но не более 50
n = TextBox2.Value
s = 0
For i = 1 To n
    Mas(i) = LD.Cells(2 + i, 2)
    s = s + Mas(i)
Next i
LR.Cells(3, 3) = s
TextBox1.Value = s
MsgBox (" Расчет выполнен")
End Sub

```

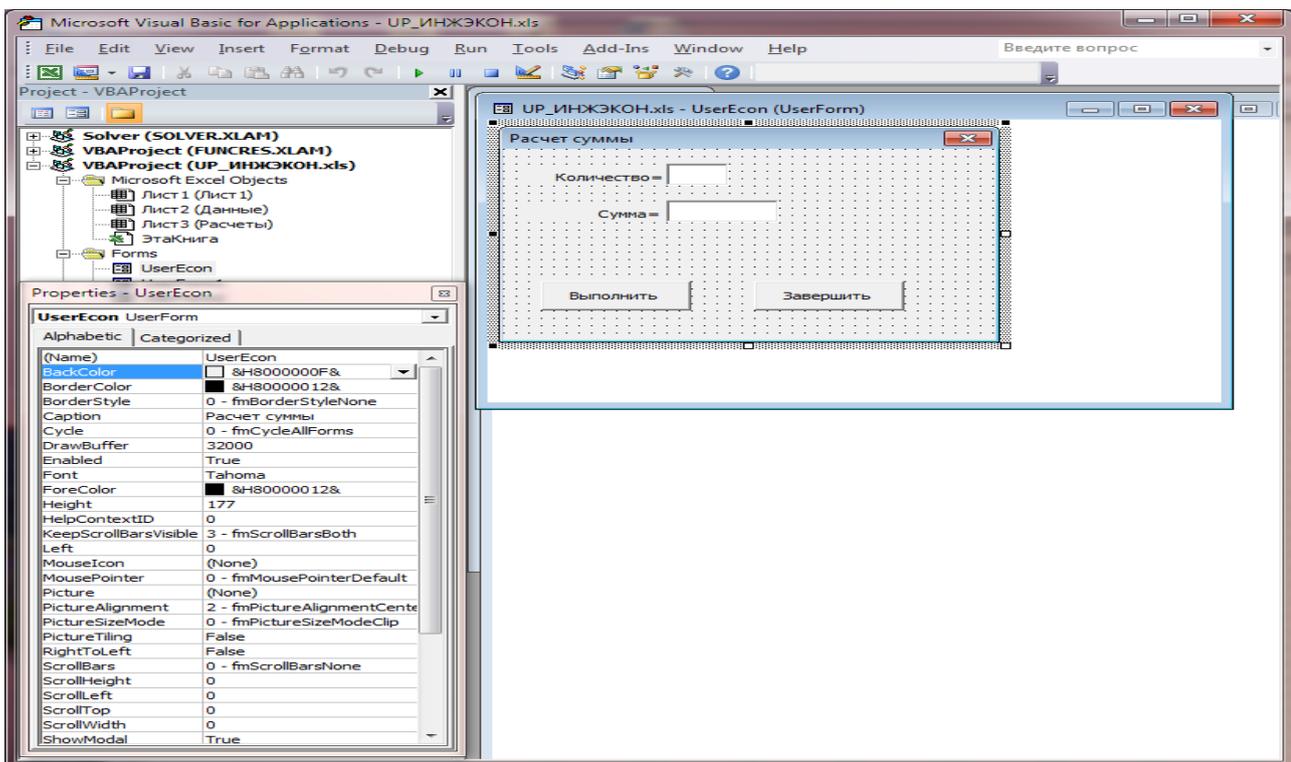


Рис. 3. Готовая форма «пользователь – программный код».

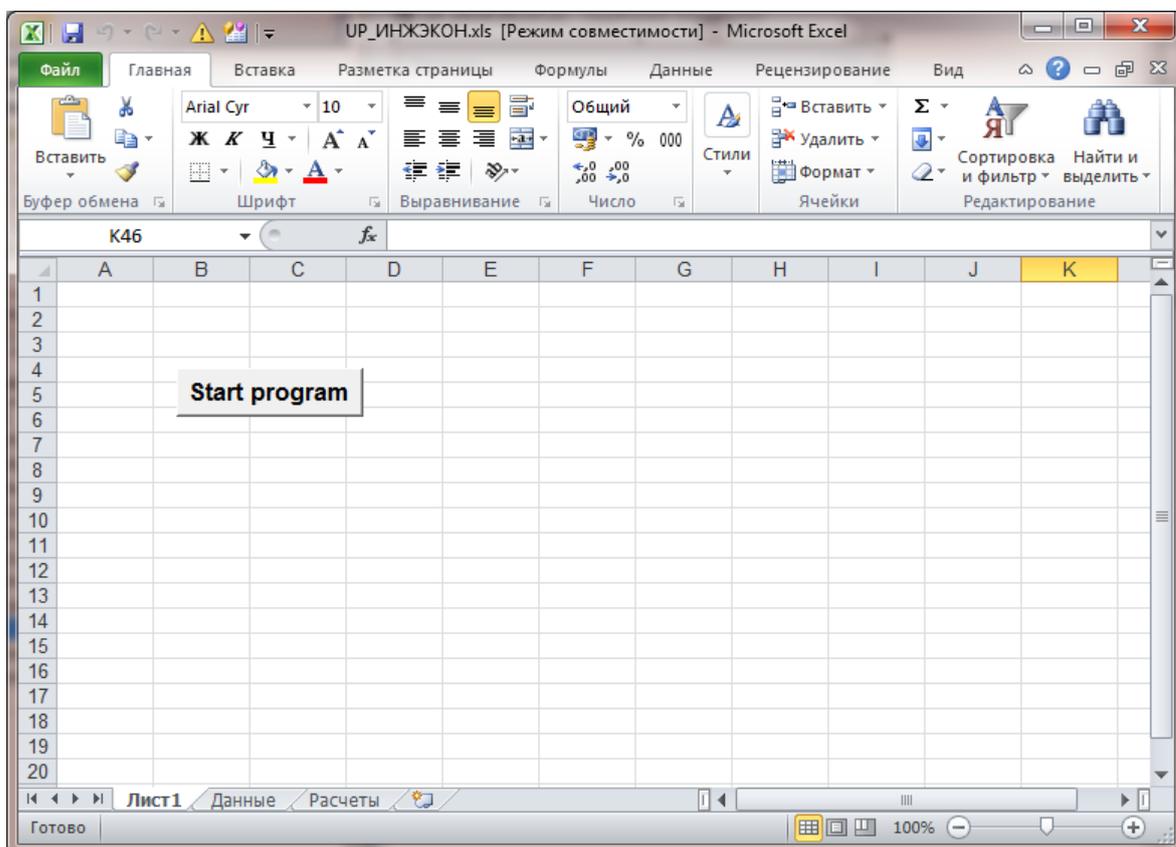


Рис.4. Запуск по команде Run (F5).

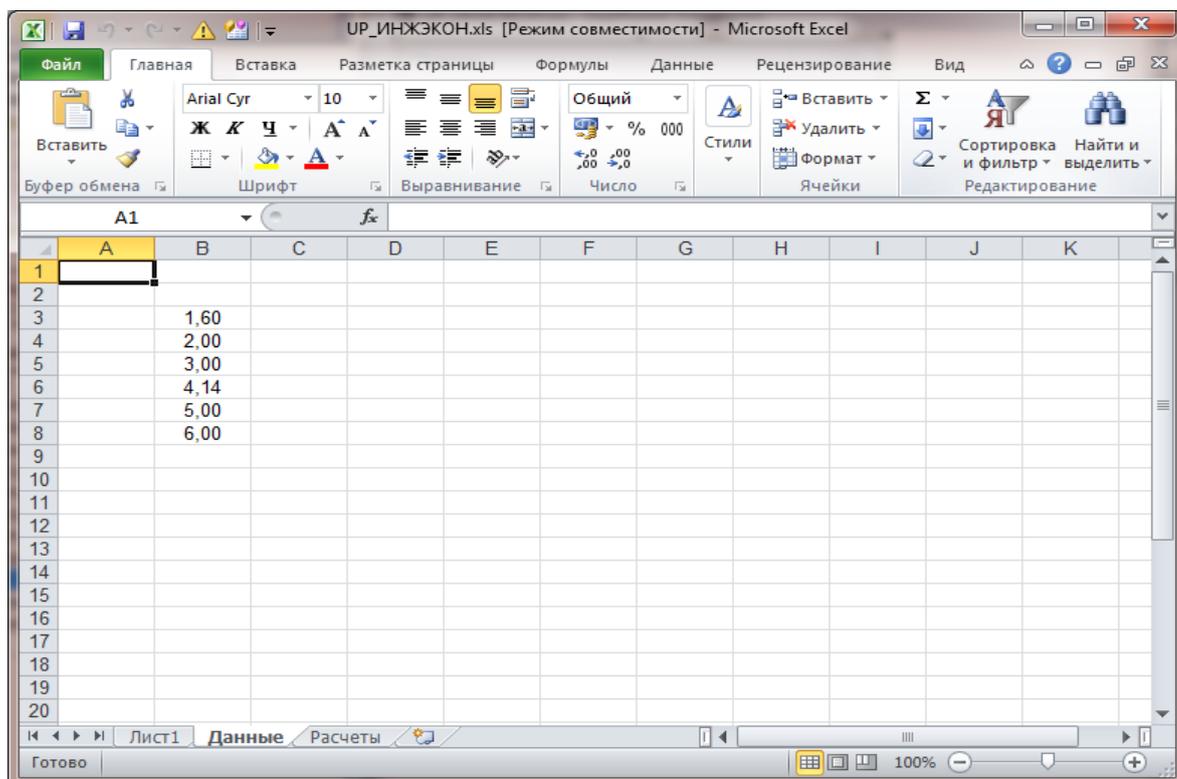


Рис. 5. Размещение числовых данных на листе «Данные».

Размещение результатов вычислений на листе «Расчеты»

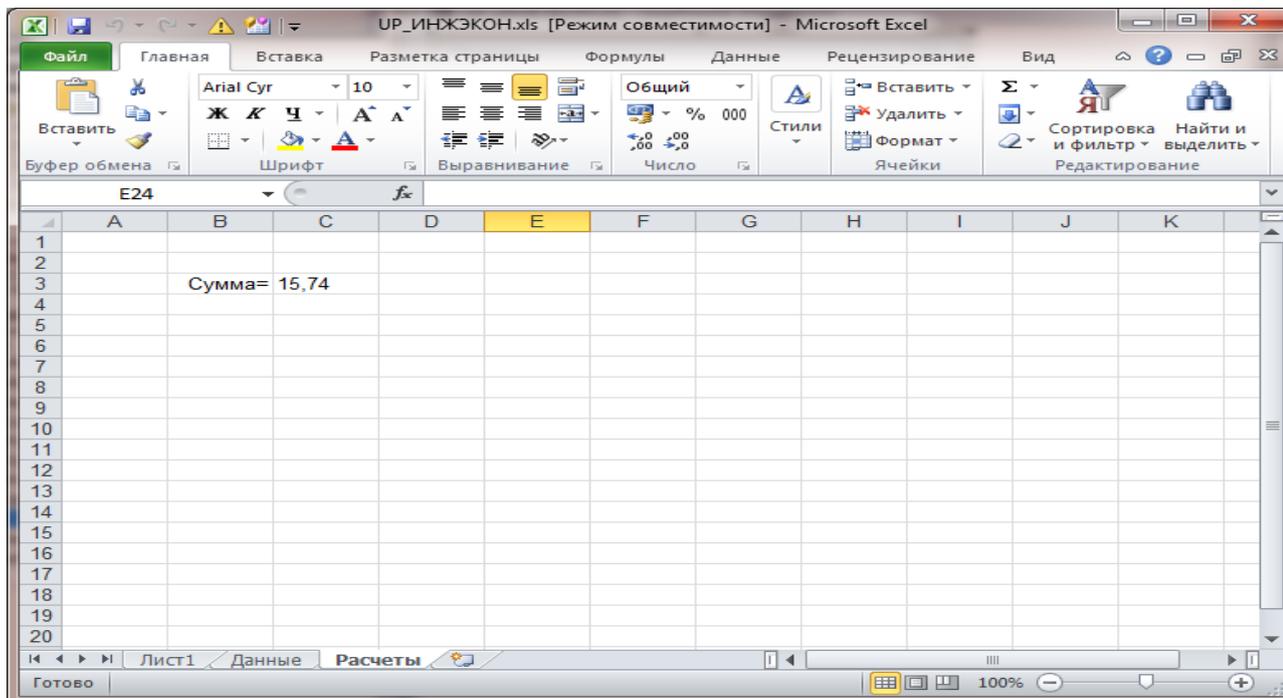


Рис. 6. Результат выполнения

Сохранение информации во внешней памяти

Файл – совокупность данных, размещенных под одним именем во внешней памяти.

MSOffice– табличный процессор – **Excel** – основной объект размещения информации – **Лист** – связь с **VBA** с помощью специальных функций, например:

SetLD = Sheets("Данные"),

здесь LD – идентификатор листа **Excel** именем “Данные”.

Файлы последовательного доступа

Последовательный файл состоит из текстовых строк ASCII-формата. Любая информация считывается или сохраняется в текстовом виде построчно. Последовательные файлы могут обрабатываться любым текстовым редактором.

Способ открытия файла с последовательным доступом (для чтения, записи или добавления) задается при вызове оператора открытия файла:

Синтаксис:

Open<имяФайла>**For**<режим>**As**<дескрипторФайла>

Здесь:

<имяФайла> - имя открываемого файла; это либо строка символов, заключенная в кавычки, либо выражение, значением которого является строка символов, что определяет маршрут к открываемому файлу;

<режим> - это режим доступа, который может иметь значения:

<дескрипторФайла> - целое число, по которому идентифицируется открываемый файл.

Параметры работы с файлами VBA

При открытии файлу ставится в соответствие канал с определенным номером. Допустимо использование номеров 1-511.

Input(Ввод) - если файл открывается для чтения из него текстовых данных.

Output(Вывод) - если файл открывается для записи в него данных, начиная с первой позиции. Вся имеющаяся в нем прежняя информация будет стерта (потеряна в результате очистки).

Append(Дополнение) – при открытии файла запись в него данных выполняется не с первой, а с конечной позиции, при этом вся имевшаяся в нем прежняя информация будет сохранена.

Все открытые файлы закрываются оператором:

Синтаксис:

Close[*списокДескрипторов*]

списокДескрипторов - это записанные через запятую дескрипторы файлов, которые должны быть закрыты. При отсутствии этого параметра все открытые файлы будут закрыты по завершению программы.

Операторы обмена информацией с внешней памятью

Для записи информации в файл используются операторы:

Print # дескрипторФайла, [*СписокЗначений*]

Write# дескрипторФайла, [*СписокЗначений*]

Здесь:

СписокЗначений - это записанные через разделитель значения (или выражения).

Оба оператора записывают данные в файл текстовыми строками. Текстовая строка - это последовательность символов, которая заканчивается символом перехода на новую текстовую

строку или символом возврата каретки (десятичные коды 13 и 10).
Текстовый файл представляет собой последовательность текстовых строк.

Логика работы операторов **Write** и **Print** различна.

Список значений просматривается последовательно, и элементы этого списка записываются в одну текстовую строку файла через запятую. Элементы типа **String** заключаются в кавычки. После записи последнего элемента записывается символ перехода на новую строку.

Пример на использование оператора **Write**

Фрагмент программы записи в текстовый файл следующих строк :

"Иванов И.И.", 2015

"Петров В.Р.", 2013

"Яблокова А.П.", 2012

"Кулиш С.П.", 2015

В каждой строке - фамилия и год рождения.

```
Private Sub Command_Click()
```

```
Dim god As Integer, fio As String
```

```
Open «дети.txt» ForOutputAs #1
```

```
For i = 1 To 4
```

```
fio = InputBox("ФИО ребенка", "Создание файла")
```

```
god = Val (InputBox(" Год рождения", "Создание файла"))
```

```
Write #1, fio, god
```

```
Next i
```

```
Close #1
```

```
EndSub
```

Пример на использование оператора **Input**

```
Public Sub ppp_out()
```

```
Dim god As Integer, fio As String
```

```
Open "student.txt" For Input As #1
```

```
For i = 1 To 4
```

```
Input #1, fio, god
```

```
MsgBox (" ФИО ребенка " &fio& " Год рождения " &god)
```

```
Next i
```

```
Close #1
```

EndSub

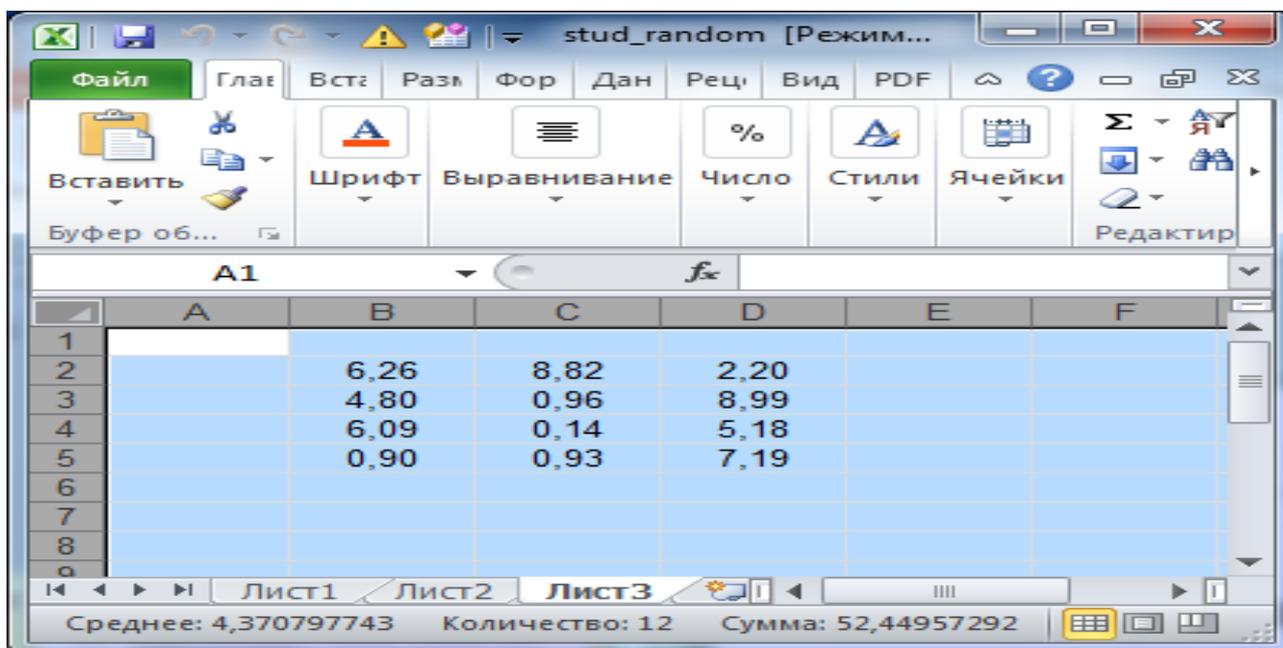
Пример решения задач с применением генератора случайных чисел

Rnd() - функция, выполняющая генерацию случайных чисел в пределах от 0 до 1.

Задача – разместить случайные значения на рабочем листе Excel. Случайные значения должны колебаться от 0 до 9. Сохранить значения с двумя знаками дробной части.

```
Public Sub Pr_random()  
Dim L3 As Object  
Dim i, j, k As Integer, v, a As Single  
Dim S, B(5, 7) As Variant  
Set L3 = Sheets(«Лист3»)  
‘v - множитель для случайных чисел  
v = 9  
For i = 1 To 4  
    For j = 1 To 3  
        B(i, j) = Rnd() * v  
        a = Round(B(i, j))  
        L3.Cells(i + 1, j + 1) = a  
        ‘MsgBox (" V=" & a)  
    Next  
Next  
End Sub
```

Результат формирования матрицы случайных чисел.



Практическое задание 1

Цель задания: приобрести навыки по обработке однотипной информации, размещенной массивами данных в оперативной памяти.

Задача:

1. разместить данные на рабочем листе **Excel**;
2. спроектировать интерфейс с помощью инструментария **ToolBox**;
2. написать программный код на **VBA** по обработке данных;
3. проверить работоспособность программы.

Варианты для индивидуальной работы:

(вариант выбирается из двух пунктов по согласованию с преподавателем)

Размерность и расположение данных на листах **Excel** произвольно.

1. **Выполнить** выборку из двумерного массива:
 - 1.1 отрицательных значений;
 - 1.2 значений больше заданной константы, константа произвольная;
 - 1.3 значений, размещенных в чётных столбцах;
 - 1.4 всех значений в заданном интервале, границы произвольные.
 - 1.5 положительных значений;
 - 1.6 значений меньше заданной константы, константа произвольная;

- 1.7 значений, размещенных в нечётных столбцах;
- 1.8 значений, размещенных на главной диагонали;
- 1.9 значений, размещенных выше главной диагонали;
- 1.10 значений, размещенных ниже главной диагонали.

2. *Вычислить*

- 2.1 количество значений больше заданной константы;
- 2.2 среднеарифметическое значений из выборки по п.1.4;
- 2.3 количество отрицательных значений из выборки по п.1.4;
- 2.4 количество значений меньше заданной константы;
- 2.5 сумму положительных значений из выборки по п.1.4;
- 2.6 сумму отрицательных значений из выборки по п.1.4;
- 2.7 количество значений больше заданной константы из выборки по п.1.4;
- 2.8 среднеарифметическое всех значений массива;
- 2.9 количество отрицательных значений из всего массива;
- 2.10 сумму положительных значений всего массива.

Пример программного кода по практическому заданию1:

Условие задачи:

1. Разместить массив числовых значений размерности 3 на 7 на **Листе1**.
2. Выполнить выборку положительных значений.
3. Разместить выборку положительных значений на **Листе2**.
4. Вычислить среднеарифметическое положительных значений.

DimL1, L2AsObject - описательная конструкция объявляет рабочие листы **Excel** как объекты.

SetL1 = Sheets("Лист1") - исполнительная конструкция – функция **Set** устанавливает связь формального имени листа с фактическим (физическим).

L1.Cells(i + 2, j + 3) – уточненное имя ячейки листа **Excel**, где

i + 2 - номер строки листа **Excel**;

j + 3 - номер столбца листа **Excel**;

i, j - текущие номера (счетчики).

Исходный текст программы

PublicSubprimer()

Dim L1, L2 As Object

```

Dim i, j, k As Integer
Dim S, B(3, 7) As Variant
Set L1 = Sheets("Лист1") : Set L2 = Sheets("Лист2")
For i = 1 To 3
For j = 1 To 7
B(i, j) = L1.Cells(i + 2, j + 3)
Next j
Next i
S = 0: k = 0
For i = 1 To 3
For j = 1 To 7
If B(i, j) > 0 Then
    S = S + B(i, j)
    L2.Cells(7, k + 1) = B(i, j)
    k = k + 1
End If
Next j
Next i
MsgBox ("Среднеезн.=" &S / k& " К=" &k)
End Sub

```

Форму пользователя для обмена информацией (ввода и вывода данных) предлагается разработать самостоятельно, пользуясь теоретическими сведениями и конкретным примером, изложенным выше.

Практическое задание 2

Цель работы: закрепить практические навыки в реализации расчетных задач небольших по объему, но часто используемых в профессиональной деятельности на базе интегрированной среды **VBA**.

Задача:

Разработать форму пользователя и написать программный код на заданную тему по выбранному варианту.

С методиками расчета можно познакомиться в учебном пособии [3].

Варианты для индивидуальной работы:

2.1 Тема: *расчет дохода по депозиту.*

Определить сумму в конце срока, если простые проценты начисляются в конце каждого года и в конце каждого квартала.

2.2. Тема: *расчет кредита.*

Определить сумму долга с ежегодным начислением, с ежеквартальным начислением сложных процентов и с ежедневным начислением сложных процентов.

2.3. Тема: *расчет эффективной ставки.*

Определить эффективную ставку, если номинальная ставка $R=25\%$ применяется ежемесячно.

2.4. Тема: *расчет номинальной ставки.*

Определить размер номинальной ставки при эффективной ставке $R=28\%$, если начисление процентов производится в конце каждого полугодия.

2.5. Тема: *расчет размера налога.*

Определить размеры налога на проценты при начислении простых и сложных процентов при ставке налога $R=5\%$.

2.6. Тема: *расчет наращенной суммы с учетом инфляции.*

Определить сумму вклада и процентные деньги при начислении простых процентов со ставкой $R=12\%$ годовых и наращенную сумму с учетом темпа инфляции $h = 0,5\%$ в месяц.

2.7. Тема: *расчет наращенной суммы ренты.*

Определить наращенную сумму ренты за N лет, платежи поступают в конце каждого месяца ($p = 12$) равными долями, на которые начисляются ежеквартально ($m = 4$) сложные проценты $R=12\%$ годовых.

2.8. Тема: *расчет накопления запланированной суммы.*

Определить величину ежемесячного платежа R при условии, что проценты начисляются в конце каждого месяца по сложной ставке в 12% годовых и запланированной сумме S .

2.9. Тема: *расчет ежемесячного платежа R ренты.*

Определить величину ежемесячного платежа R ренты, современная стоимость которой за 2 года составит 300 000. Проценты начисляются в конце каждого месяца по сложной ставке в 12% годовых.

2.10. Тема: *расчет цены облигации.*

Определить цену облигации сроком гашения $G=2$ года при выплате по купону один раз в году. На рынке доходность на инвестиции с уровнем риска, соответствующим данной облигации, оценивается в $S=15\%$..

Пример программного кода по практическому заданию 2:

Условие задачи:

Тема: *расчет наращенного вклада по процентной ставке.*

Расчет суммы по обычной процентной ставке начисляется в зависимости от первоначального вклада.

Здесь следующие обозначения:

P – первоначальная сумма, т. е. сумма денег, получаемых заемщиком,

i – процентная ставка за период (десятичная дробь),

n – срок ссуды (количество периодов начисления процентов),

S – наращенную сумму, т.е. сумму денег с процентами.

По истечении срока ссуды сумма будет равна

$$S = P(1 + ni)$$

Определить: кварталные, ежемесячные и годовые процентные ставки.

The image shows a window titled "Расчет суммы" (Calculation of sum). It contains the following elements:

- Three input fields for "Сумма кредита (руб)" (Loan amount in rubles), "Кредитный период (лет)" (Loan term in years), and "Процентная ставка годовых (%)" (Annual interest rate (%)).
- A section titled "Вариант начисления сложных процентов" (Option of compounding interest) with three radio buttons: "ежегодное" (annual), "ежеквартальное" (quarterly), and "ежедневное" (daily).
- Two buttons: "Выполнить" (Execute) and "Завершить" (Finish).
- An output field labeled "Сумма долга" (Total debt) at the bottom.

Рис. 7. Проект формы пользовательского интерфейса.

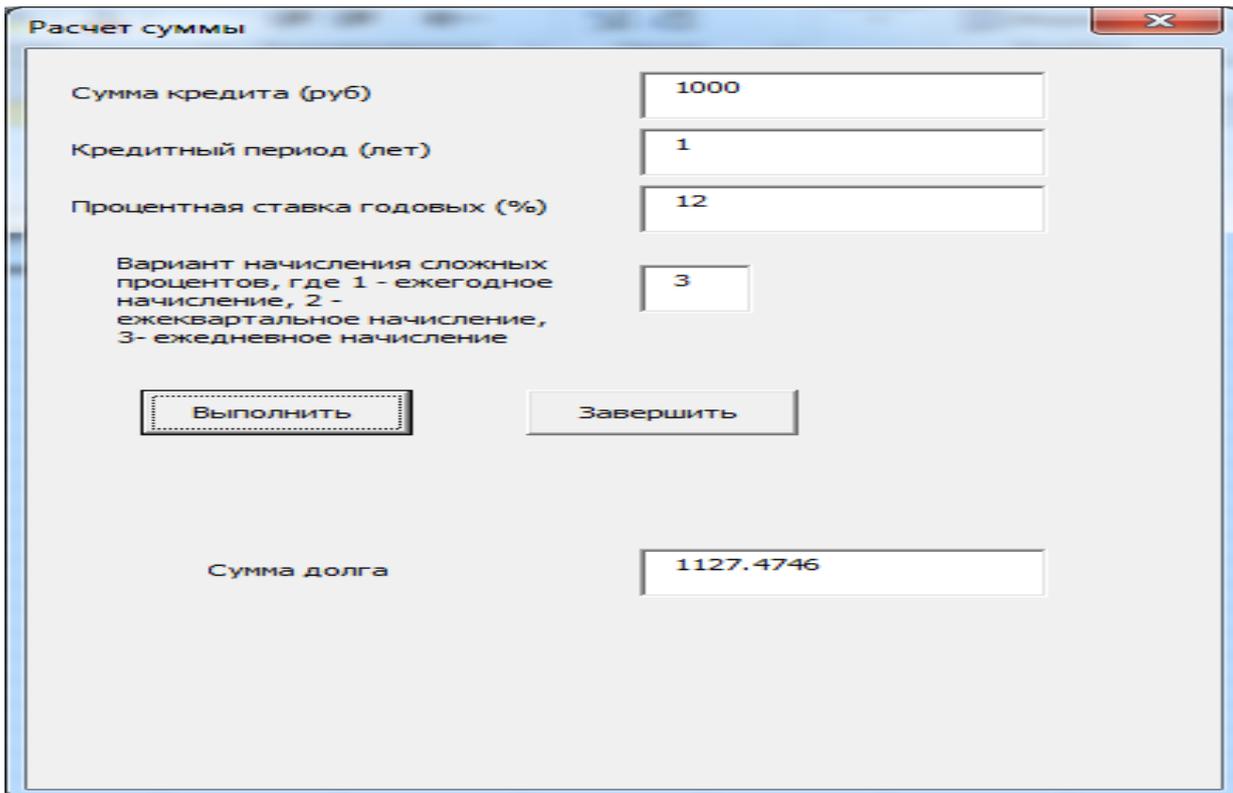
Программный код по командной кнопке «Завершить»

```
Private Sub CommandButton2_Click()  
    Unload Me  
EndSub
```

Программный код по командной кнопке «Выполнить»

```
PrivateSubCommandButton1_Click()  
' Условие проверки начисления  
Dim S As Currency  
Dim i, n, P As Long  
P = TextBox1.Value : n = TextBox2.Value : i = TextBox3.Value  
If OptionButton1.Value = True Then  
S = P * ((1 + i / 100) ^ n) : TextBox5.Value = S  
    Else  
        If OptionButton2.Value = True Then  
            S = P * ((1 + i / 400) ^ (4 * n)) : TextBox5.Value = S  
        Else  
            If OptionButton3.Value = True Then  
                S = P * ((1 + i / 36500) ^ (365 * n)) : TextBox5.Value = S  
            Else: MsgBox(«Некорректный вариант начисления»)  
        End If  
    End If  
    End If  
End Sub
```

Проверка работоспособности программы



Расчет суммы

Сумма кредита (руб)

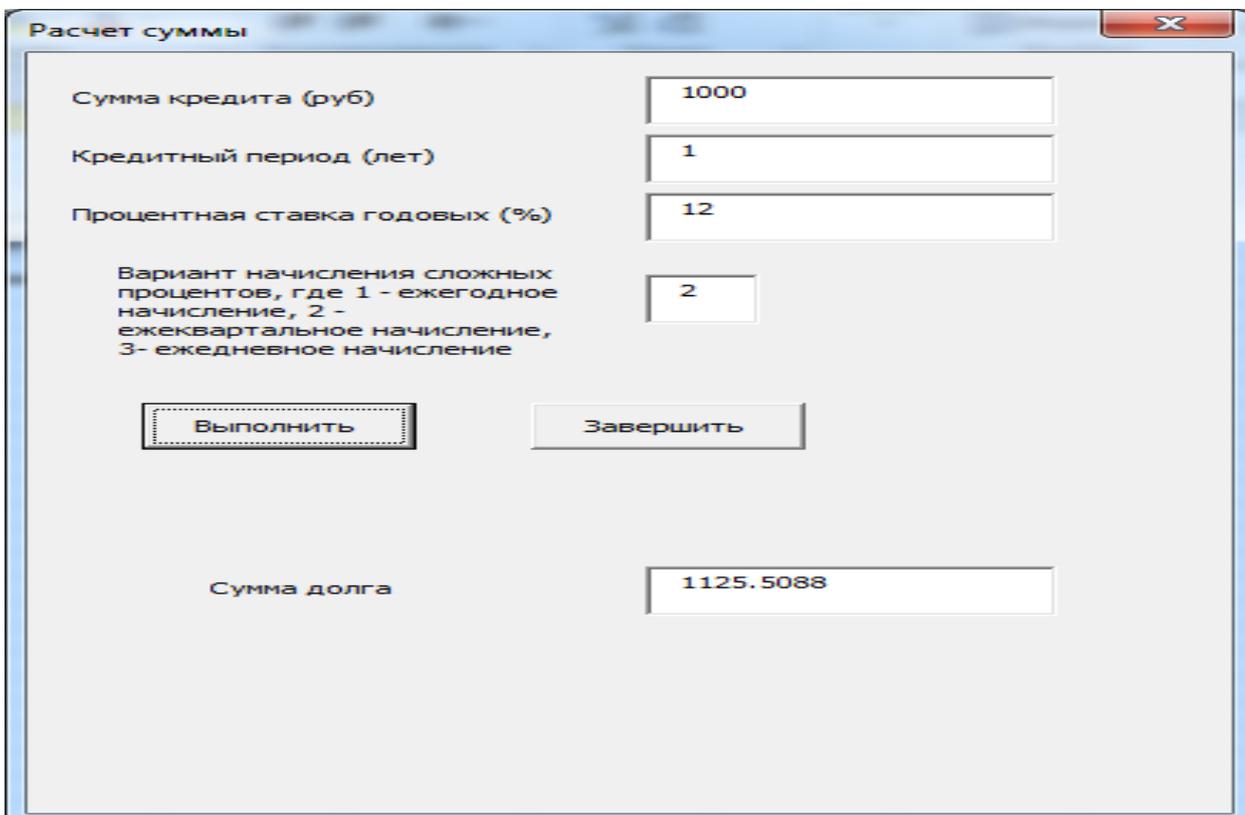
Кредитный период (лет)

Процентная ставка годовых (%)

Вариант начисления сложных процентов, где 1 - ежегодное начисление, 2 - ежеквартальное начисление, 3 - ежедневное начисление

Сумма долга

Рис. 8. Ежегодное начисление.



Расчет суммы

Сумма кредита (руб)

Кредитный период (лет)

Процентная ставка годовых (%)

Вариант начисления сложных процентов, где 1 - ежегодное начисление, 2 - ежеквартальное начисление, 3 - ежедневное начисление

Сумма долга

Рис. 9. Ежеквартальное начисление

Расчет суммы

Сумма кредита (руб)

Кредитный период (лет)

Процентная ставка годовых (%)

Вариант начисления сложных процентов, где 1 - ежегодное начисление, 2 - ежеквартальное начисление, 3 - ежедневное начисление

Сумма долга

Рис.10. Ежедневное начисление.

Результаты расчетов по разным методикам сравнить и выбрать наиболее подходящий вариант дохода. Результаты расчетов по разным методикам сравнить, проанализировать и выбрать наиболее подходящий вариант дохода. Из выполненной работы следует вывод - часто используемые расчетные небольшие задачи в автоматизированном режиме могут существенно упростить каждодневную профессиональную деятельность.

Практическое задание 3

Цель работы: приобрести практические навыки в реализации задач по частичной автоматизации обработки информации с использованием средств интегрированной среды **VBA**.

Задача:

Разработать форму пользователя и написать программный код на заданную тему по выбранному варианту.

Варианты для индивидуальной работы:

Вариант 1

Создать Windows-приложение "Формирование программы строительно-монтажных работ". Предусмотреть:

1. Создание входных форм:

Смета работ (Код объекта, Наименование объекта, Код и Объем работы);

Справочник расценок (Код работы, Наименование работы, Расценка)

2. На основе входных данных построить выходной документ:

Стоимость работ (Код объекта, Итоговая стоимость работ).

3. Выполнить расчет итоговой стоимости работ по заданному Коду объекта.

Вариант 2

Создать Windows-приложение "*Формирование счетов на оплату в автосервисе*". Предусмотреть:

1. Создание входных форм:

Справочник видов работ (Код работы, Марка автомобиля, Наименование работы, Стоимость работы);

Заказы (Дата, Номер заказа, Клиент, Марка автомобиля, Код работы).

2. На основе входных данных построить выходной документ:

Счет (Код объекта, Итоговая стоимость работ).

3. Выполнить расчет стоимости заказа по заданному Номеру заказа.

Вариант 3

Создать Windows-приложение "*Учет оплаты заказов*". Предусмотреть:

1. Создание входных форм:

Справочник услуг (Код услуги, Наименование услуги, Стоимость услуги);

Клиенты (Код клиента, Наименование клиента, Адрес, Телефон).

2. На основе входных данных построить выходной документ:

Заказы (Номер заказа, Дата заказа, Код клиента, Код услуги, Стоимость услуги, Форма оплаты).

3. Выполнить расчет суммы оплаты заказов по заданной Форме оплаты.

Вариант 4

Создать Windows-приложение "Учет торговых заказов".
Предусмотреть:

1. Создание входных форм:

Каталог товаров (Код товара, Наименование товара, Цена товара);

Клиенты (Код клиента, Наименование клиента, Адрес, Телефон).

2. На основе входных данных построить выходной документ:

Заказы (Номер заказа, Дата заказа, Код клиента, Код товара, Количество товара).

3. Выполнить расчет стоимости заказа по заданному Номеру заказа.

Вариант 5

Создать Windows-приложение "Формирование реестра заказов". Предусмотреть:

1. Создание входных форм:

Каталог услуг(Код услуги, Наименование услуги, Стоимость услуги);

Клиенты (Код клиента, ФИО клиента, Адрес, Телефон).

2. На основе входных данных построить выходной документ:

Заказы (Номер заказа, Дата заказа, Код клиента, Код услуги, Стоимость услуги).

3. Выполнить расчет общей стоимости заказа по заданному Номеру заказа.

Вариант 6

Создать Windows-приложение "Учет движения продукции на складе". Предусмотреть:

1. Создание входных форм:

Движение (Код продукции, Наименование, Остаток на начало);

Накладные на поступление (Номер накладной, Код продукции, Количество поступлений);

Требования на выдачу (Номер требования, Код продукции, Количество выдано).

2. На основе входных данных построить выходной документ:

Учет остатков(Код продукции, Остаток на конец).

3. Выполнить расчет остатков на начало и конец срока по заданному Коду продукции.

Вариант 7

Создать Windows-приложение "*Расчет зарплаты к выдаче работникам*".

Предусмотреть:

1. Создание входных форм:

Справочник работников(Табельный номер, ФИО работника, Код отдела, Количество льгот);

Начисления (Табельный номер, начислено).

2. На основе входных данных построить выходной документ:

Удержания (Табельный номер, Удержания в ПФ, Удержания ПН).

3. Выполнить расчет зарплаты к выдаче по заданному Табельному номеру.

Вариант 8

Расчет удержаний с зарплаты

Создать Windows-приложение "*Расчет удержаний с зарплаты*". Предусмотреть:

1. Создание входных форм:

Справочник работников (Табельный номер, ФИО работника, Код отдела, Количество льгот);

Начисления (Табельный номер, начислено).

2. На основе входных данных построить выходной документ:

Удержания (Табельный номер, Удержания в ПФ, Удержания ПН).

3. Выполнить расчет общей суммы удержаний по заданному Табельному номеру.

Вариант 9

Создать Windows-приложение "*Формирование плана выпуска продукции*". Предусмотреть:

1. Создание входных форм:

Справочник продукции(Код, Наименование продукции, Цена сборки);

Справочник деталей (Код детали, Наименование, Цена);

Состав продукции(Код продукции, Код детали, Количество деталей в продукции).

2. На основе входных данных и планового количества продукции построить выходной документ:

План выпуска продукции (Код, Стоимость выпуска продукции).

3. Выполнить расчет Стоимости выпуска заданной продукции.

Вариант 10

Создать Windows-приложение "Учет выполнения работ в автосервисе". Предусмотреть:

1. Создание входных форм:

Справочник видов работ (Код работы, Марка автомобиля, Наименование работы, Код исполнителя);

Справочник исполнителей работ (Код исполнителя, ФИО);

Заказы (Дата, Номер заказа, ФИО клиента, Марка автомобиля, Наименование работы, Код исполнителя).

2. На основе входных данных и заданных Номеров заказа построить выходной документ:

Выполнение работ (Код исполнителя, Номер заказа, Марка автомобиля, Стоимость заказа).

3. Выполнить расчет стоимости заказа по заданному Номеру заказа.

Пример программного кода по практическому заданию 3:

Условие задачи: *рассчитать за отработанное время зарплату работников - учесть разряд и коэффициент сложности.*

Первая таблица содержит информацию о работающих (Табельный номер, Фамилия, Разряд);

Вторая - справочник по разрядам (Разряд, Оклад),

Третья - сведения о начислении зарплаты (Табельный номер, Фамилия, Коэффициент сложности - отработанное время, Начислено).

	А	В	С
1	Таб.номер	Фамилия	Разряд
2	1111	Иванов	12
3	1122	Петров	15
4	1222	Сидоров	13

Рис.11. Информация о работающих сотрудниках.

	А	В	С	Д
1	Таб.номер	Фамилия	Коэффициент	Начислено
2	1111	Иванов	5	600
3	1122	Петров	8	720
4	1222	Сидоров	12	630
5	2222	Марков	7	560

Рис. 12. Сведения о начислении зарплаты.

	А	В
1	Разряд	Оклад
2	10	400
3	11	500
4	12	600
5	13	700
6	14	800
7	15	900

Рис. 13. Информационный справочник о разрядах.

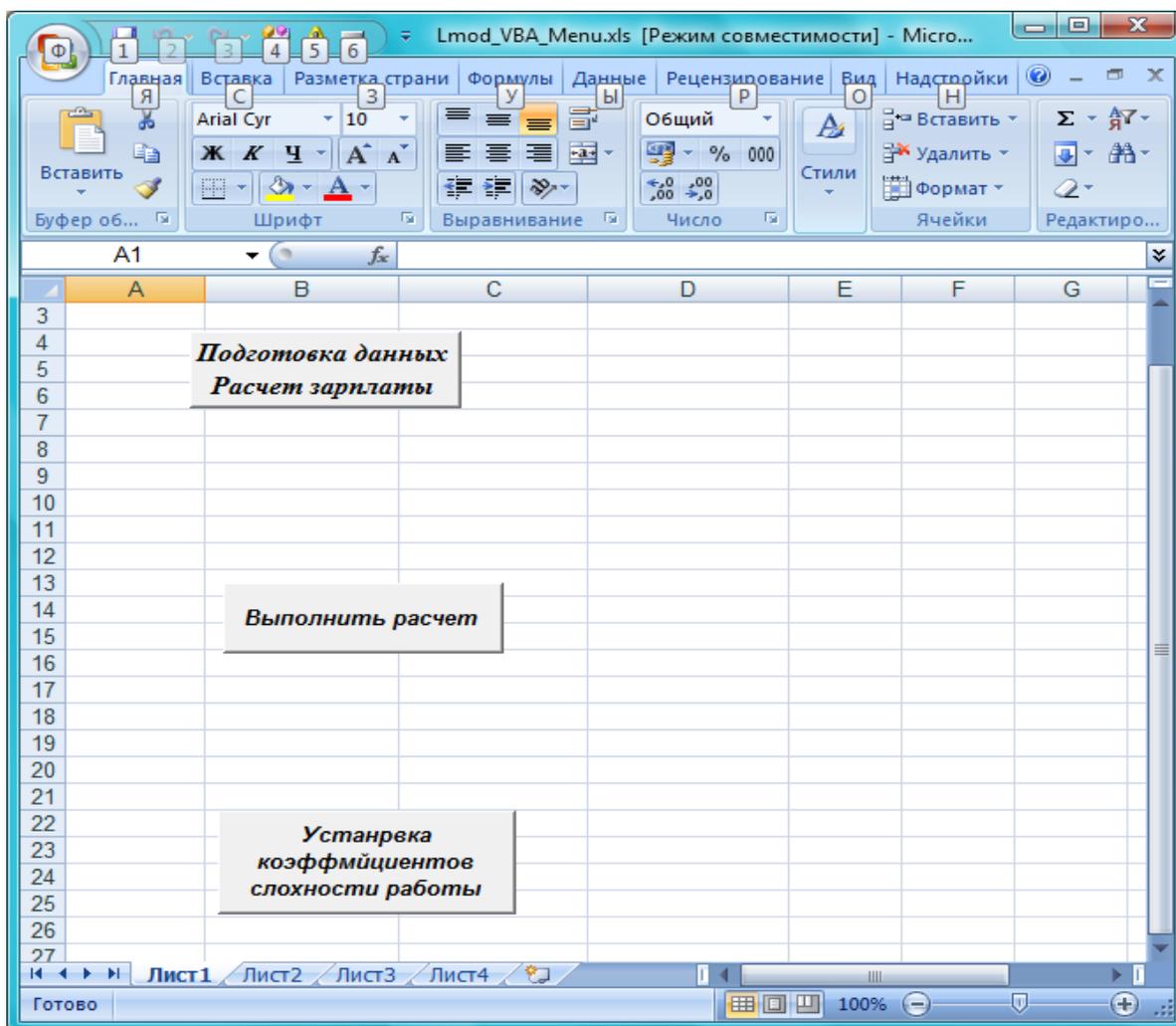


Рис. 14. Командные кнопки управления.

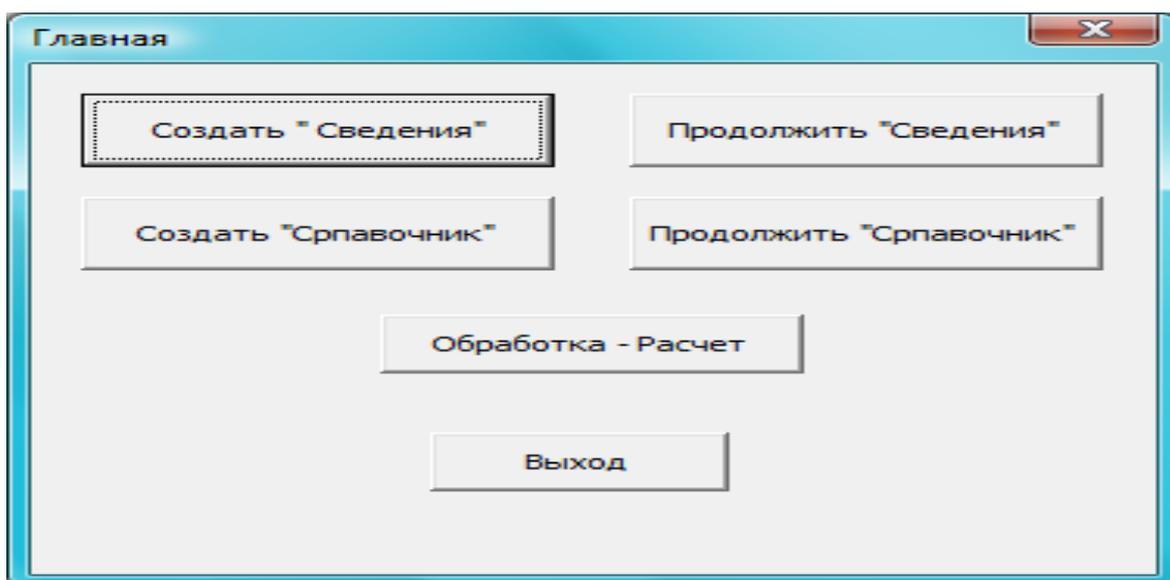


Рис. 14. Главное меню вызывается по запуску командной кнопки «Подготовка данных. Расчет зарплаты».

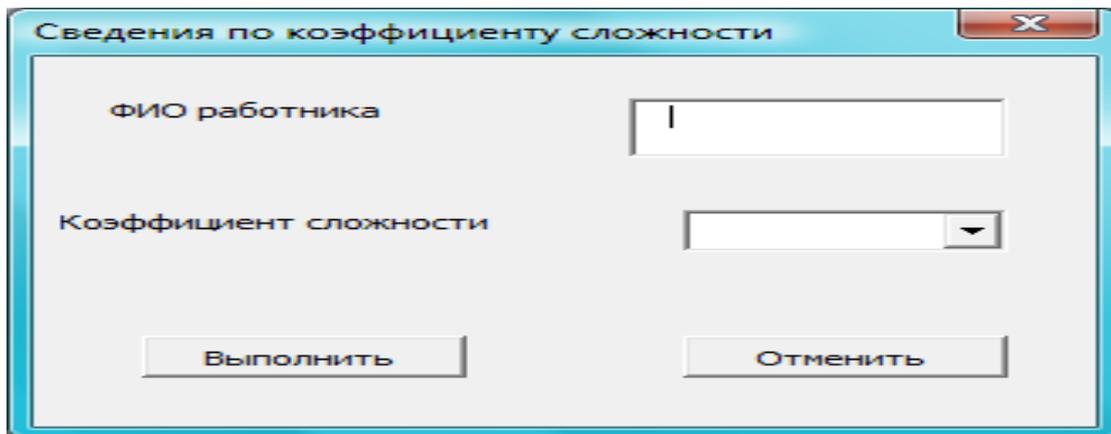


Рис. 15. Меню «Сведения по коэффициенту сложности» вызывается по запуску командной кнопки «*Установка коэффициентов сложности работы*».

Текст программного модуля - «Информация о работающих сотрудниках»

Первоначальное занесение данных

```

Private Sub cm_1_Click()
Sheets(2).Activate
Range("A:D").Clear
ActiveSheet.Cells(1, 1) = "Таб.номер"
ActiveSheet.Cells(1, 2) = "Фамилия"
ActiveSheet.Cells(1, 3) = "Разряд"
Cells(1, 1).Activate
Сведения.Show
End Sub

```

Текст программного модуля - «Информация о работающих сотрудниках»

Добавление данных

```

Private Sub CommandButton1_Click()
Sheets(2).Activate
Сведения.Show
EndSub

```

Текст программного модуля - « Информационный справочник о разрядах»

Первоначальное занесение данных.

```
Private Sub cm_2_Click()  
Sheets(3).Activate  
Range("A:B").Clear  
ActiveSheet.Cells(1, 1) = "Разряд"  
ActiveSheet.Cells(1, 2) = "Оклад"  
Cells(1, 1).Activate  
Справочник.Show  
End Sub
```

Добавление данных.

```
Private Sub CommandButton2_Click()  
Sheets(3).Activate  
Справочник.Show  
End Sub
```

Текст программного модуля - Запуск на расчет зарплаты

```
Private Sub cm_3_Click()  
Calculation  
End Sub
```

Возврат в главное меню или отмена

```
Private Sub cm_4_Click()  
End  
End Sub
```

Текст программного модуля - «Сведения о начислении зарплаты»

```
Sub Calculation()  
Dim v As Integer  
Sheets(4).Activate  
Range("A:D").Clear  
ActiveSheet.Cells(1, 1) = "Таб. номер"
```

```

ActiveSheet.Cells(1, 2) = "Фамилия"
ActiveSheet.Cells(1, 3) = "Коэффициент"
ActiveSheet.Cells(1, 4) = "Начислено"
i = 2: J = 2
Do While Sheets(2).Cells(J, 1) <> ""
ActiveSheet.Cells(i, 1) = Sheets(2).Cells(J, 1)
ActiveSheet.Cells(i, 2) = Sheets(2).Cells(J, 2)
v = InputBox("Ввод значения коэффициента")
ActiveSheet.Cells(i, 3) = v
k = 2
Do While Sheets(3).Cells(k, 1) <> ""
If Sheets(2).Cells(J, 3) = Sheets(3).Cells(k, 1) Then
ok = Sheets(3).Cells(k, 2)
End If
k = k + 1
Loop
ActiveSheet.Cells(i, 4) = ActiveSheet.Cells(i, 3) * ok
J = J + 1
i = i + 1
Loop
End Sub

```

Выполнение расчета зарплаты возможно после установки сложности исполняемых работ.

Проверка работоспособности программы

Тестовый пример – результат выполнения.

Таб.номер	Фамилия	Коэффициент	Начислено
112	Демин	11	9900
115	Ванин	3	2700
116	Попов	3	1800
125	Антонов	4	2800
121	Фролов	8	4800
204	Носов	5	2500
209	Воронов	6	3000
210	Сафронов	9	4500
211	Иванов	6	6000
215	Яковлев	7	6300

Лист 4. Состояние после выполнения

Таб.номер	Фамилия	Разряд	Коэффициент
112	Демин	10	11
115	Ванин	10	3
116	Попов	11	3
125	Антонов	9	4
121	Фролов	11	8
204	Носов	7	5
209	Воронов	7	6
210	Сафронов	7	9
211	Иванов	8	6
215	Яковлев	12	7

Лист 2. Состояние после выполнения

Разряд	Оклад
4	100
11	600
10	900
9	700
7	500
8	1000
2	100
3	200
5	300
6	400
12	900

Лист 3. Состояние после выполнения

Расчетные формулы условны. В данном примере показаны возможности интегрированной среды VBA для реализации подобных задач.

Библиографический список

1. Рачева, Н.В., Сибирев В.Н.: Алгоритмирование и программирование: учебно-методический комплекс / сост. Н.В.Рачева, В.Н.Сибирев. – СПб.: Изд-во СЗТУ, 2009. – 81 с.
2. Якушева, Н.М. Введение в программирование на языке VisualBasic.NET: учебное пособие. – М.: Финансы и статистика, 2006. – 320 с.
3. Боброва, Л.В.: Финансовая математика: учебное пособие/ Л.В. Боброва, Ю.С. Романова, Е.А. Рыбакова – СПб.: НОИ, 2015. - 51 с.

СОДЕРЖАНИЕ

1. Теоретические сведения.....	3
2. Практическое задание 1.....	14
3. Практическое задание 2	16
4. Практическое задание 3	21
5.Библиографический список.....	31