

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АКАДЕМИЧЕСКИЙ КОЛЛЕДЖ

ОТЧЕТ
ПО УЧЕБНОЙ ПРАКТИКЕ
по профессиональному модулю
ПМ.01 Осуществление интеграции программных модулей
программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование
период с «15» мая по «03» июня 2023 года

Наименование предприятия: ФГБОУ ВО «ВВГУ», Академический колледж,
г. Владивосток

Студент гр. СО-ИП-21-1

Руководитель практики
от предприятия

Отчет защищен:
с оценкой



Руководитель
практики от ОО



О.О. Бабкина



Д.А. Атабаева

Д.А. Атабаева

Владивосток 2023

Содержание

Введение _____	3
1 Описание предметной области _____	4
2 Разработка программного модуля _____	8
2.1 Разработка алгоритма и программирование модуля _____	8
2.2 Инспектирование программного модуля на предмет соответствия стандартам кодирования _____	30
Заключение _____	32
Список используемых источников _____	33

Введение

Основной целью учебной практики является повышение профессиональной компетенции по профессиональному модулю ПМ.01 «Осуществление интеграции программных модулей», а также закрепление, расширение и углубление полученных теоретических знаний, приобретение практических навыков самостоятельной работы и выработку умений применять их на практике.

Задачами практики являются:

- ознакомление с документацией и перечнем заданий;
 - разработка программного модуля:
 - 1) проведение анализа требований к программным модулям по предложенной документации учебной практики;
 - 2) разработка алгоритма и структуры данных;
 - 3) программирование модуля;
 - 4) шлифовка текста модуля;
 - 5) проверка модуля;
 - 6) транслирование модуля;
 - разработка текстового набора и сценария программного средства для программного модуля;
 - инспектирование разработанных программных модулей на предмет соответствия стандартам кодирования:
 - 1) инспектирование программного кода на предмет соответствия стандартам кодирования, разработанного программного модуля;
 - 2) инспектирование функционального тестирования текстовых наборов и сценариев программных средств;
 - 3) инспектирование тестирования интеграции текстовых наборов и сценариев программных средств;
 - обобщение материалов практики;
 - оформление отчета и необходимой документации по практике.
- Практику проходила в ФГБОУ ВО «ВВГУ», Академический колледж, г. Владивосток.

1 Описание предметной области

Согласно выбранному варианту, следовало выполнить следующие задания:

1. Написать программу, которая находит наименьшее число во вводимой с клавиатуры последовательности, состоящей из 20 действительных чисел.

2. При помощи датчика случайных чисел вводится последовательность $\{a_i\}$, состоящая из N (случайное число от 7 до 30) целых положительных случайных трехзначных чисел. Из последовательности $\{a_i\}$ получить последовательность $\{b_i\}$, переставив в элементах a_i первую и вторую цифры. Обе последовательности вывести в соседние столбцы рабочего листа.

3. В одномерный массив A размерности N при помощи датчика случайных чисел вводятся действительные числа в диапазоне от 900 до 4500 (N ввести с клавиатуры). Вывести их в строку рабочего листа. Все те числа, в которых целая часть кратна 4, выделить цветом.

4. Написать оптимальную (по числу арифметических операций) программу для вычисления приведенной ниже функции:

$$F(x) = \frac{\sin(x^{1,2})}{2^2} - \frac{\sin(x^{1,4})}{2^4} + \frac{\sin(x^{1,6})}{2^6} - \dots + \frac{\sin(x^4)}{2^{30}} - \frac{\sin(x^{4,2})}{2^{32}}.$$

5. Написать программу для вычисления функции:

$$y = \begin{cases} 0, & x < -1 \\ \cos(\pi x), & x \in [-1, 0) \\ x^2 + 1, & x \in [0, 2) \\ 7 - x, & x \in [2, 7) \\ 0, & x \geq 7 \end{cases}.$$

Программа должна автоматически строить график данной функции на произвольном отрезке $[a;b]$ (a и b вводятся с клавиатуры).

6. В массив A считать 100 действительных чисел, находящихся в первом столбце рабочего листа. Отсортировать элементы, стоящие на нечетных местах по возрастанию, а на четных местах по убыванию элементов. Полученный массив записать во второй столбец. Выделить разным цветом ячейки, соответствующие возрастающим и убывающим подмассивам.

7. При помощи датчика случайных чисел заполнить целочисленную квадратную матрицу A порядка N (N ввести с клавиатуры). Столбцы матрицы A , имеющие нечетную сумму элементов, отсортировать по убыванию элементов. Вывести сумму элементов столбцов, а также матрицу A до и после сортировки. Все элементы отсортированных столбцов выделить цветом.

8. Во введенном с клавиатуры предложении поменять порядок слов на обратный, т.е. первое слово поменять с последним, второе – с предпоследним и так до среднего слова.

9. Написать логическую функцию, принимающую значения True, если в целом числе, являющимся единственным аргументом этой функции, сумма цифр – простое число.

10. В первых 10 столбцах рабочего листа находятся сведения о сотрудниках фирмы. Причем в первых трех столбцах рабочего листа записаны: фамилия, имя и отчество. Необходимо написать программу, считывающую эти данные и сортирующую этот список по возрастанию количества букв, содержащихся в первых трех столбцах. Имена полей и их содержимое придумать самостоятельно. Результаты работы программы вывести на другой рабочий лист.

11. При помощи датчика случайных чисел получить и вывести в столбец рабочего листа Excel N (N вводится с клавиатуры) целых чисел типа Long. В соседний столбец вывести представление соответствующего числа в дополнительном коде шестнадцатеричной системы счисления.

12. Написать программу, которая умножает сверхбольшие целые числа, имеющие более 100 цифр. Определить сумму цифр в сверхбольшом числе $D=A+B+C$. Построить гистограмму частоты появления цифр в полученном сверхбольшом числе.

13. Считать с первого столбца рабочего листа N (N вводится с клавиатуры) вещественных чисел типа Single. Перевести числа в шестнадцатеричный формат и вывести в соседний столбец. В шестнадцатеричном представлении числа поменять 1-ую цифру с 8-ой, 2-ую с 7-ой, 3-ю с 6-ой, 4-ю с 5-ой и т.д. Полученное число перевести в десятичную систему счисления и вывести в 3 столбец рабочего листа.

14. Написать программу, которая считывает в одномерный массив A целые числа, записанные в столбце рабочего листа, начиная с выделенной ячейки. Ячейки, в которых находятся числа, в разложении которых на простые множители, получается наибольшее число множителей, выделить цветом, а в соседний столбец вывести разложение таких чисел на множители.

15. Разработать игру «Морской бой» на языке VBA.

Задания с 1 по 8 и с 10 по 15 выполнены в программе Microsoft Excel, а задание 9 в программе Microsoft Word.

Microsoft Excel и Microsoft Word – это две популярные программы, входящие в пакет Microsoft Office, предназначенные для работы с таблицами и текстовыми документами соответственно. Вот краткое описание каждой из них:

Microsoft Excel:

– Excel – это электронная таблица, которая позволяет пользователям создавать, редактировать и анализировать данные в виде таблиц.

- она предлагает множество функций и возможностей для работы с числовыми данными, формулами, графиками и диаграммами.

- Excel позволяет пользователям выполнять математические операции, финансовый анализ, создавать бюджеты, таблицы учета и другие задачи, связанные с обработкой числовых данных.

- программа имеет гибкую структуру, поддерживает автоматизацию с помощью макросов и языка программирования VBA.

Microsoft Word:

- Word – это текстовый процессор, который предназначен для создания и редактирования текстовых документов, таких как письма, отчеты, резюме и другие.

- он обладает широкими возможностями для форматирования текста, включая выбор шрифтов, размеров, стилей, выравнивания, межстрочного интервала и т. д.

- Word позволяет добавлять иллюстрации, таблицы, графики, гиперссылки и другие элементы в текстовый документ.

- программа также предлагает функции проверки орфографии и грамматики, создания содержания, шаблонов документов и других инструментов для удобной работы с текстом.

Как часть пакета Microsoft Office, Excel и Word интегрируются с другими приложениями, такими как PowerPoint и Outlook, и предлагают возможности обмена данными между ними. Они являются популярными и широко используемыми инструментами в бизнесе, образовании и повседневной жизни для обработки информации и создания профессиональных документов.

VBA (Visual Basic for Applications) – это язык программирования, разработанный компанией Microsoft, который используется для автоматизации и расширения возможностей приложений Microsoft Office, таких как Excel, Word, PowerPoint и других. Ниже представлено краткое описание VBA:

- VBA – это диалект языка программирования Visual Basic, который был адаптирован для использования в приложениях Microsoft Office.

- он предоставляет возможность создания макросов, скриптов и пользовательских программ для автоматизации повторяющихся задач, обработки данных и расширения функциональности приложений Office.

- VBA позволяет взаимодействовать с объектами приложений Office, такими как рабочие книги Excel, документы Word или презентации PowerPoint, и выполнять различные операции с данными, форматированием, графиками, таблицами и другими элементами.

- он предлагает широкий набор встроенных функций и методов для работы с данными, строками, числами, массивами, файлами, пользовательскими формами и другими компонентами Office.

- VBA также поддерживает использование условных операторов, циклов, процедур, функций, обработки ошибок и других основных конструкций языка программирования.

- объектно-ориентированное программирование (ООП): VBA поддерживает основные принципы ООП, такие как инкапсуляция, наследование и полиморфизм. Он позволяет создавать классы, объекты и модули, а также взаимодействовать с объектами, свойствами и методами приложений Microsoft Office.

- пользовательский код на VBA может быть написан непосредственно в модулях приложений Office или встроен в объекты форм или элементы управления пользовательского интерфейса.

- VBA обеспечивает доступ к функциям COM (Component Object Model) и другим API, что позволяет взаимодействовать с внешними приложениями, базами данных, файловыми системами и другими системными ресурсами.

- VBA является мощным инструментом для настройки и автоматизации приложений Office, позволяя пользователям создавать настраиваемые решения, улучшать производительность и эффективность работы, а также расширять функциональность приложений в соответствии с индивидуальными потребностями.

- совместимость со сторонними библиотеками: VBA позволяет использовать сторонние библиотеки и компоненты, расширяя возможности языка. Это позволяет интегрировать функциональность сторонних продуктов и расширять возможности разработки в VBA.

- отладка и тестирование: VBA предоставляет инструменты для отладки и тестирования кода. Вы можете устанавливать точки останова, выполнять код по шагам, проверять значения переменных и исправлять ошибки в процессе разработки [1].

2 Разработка программного модуля

2.1 Разработка алгоритма и программирование модуля

Согласно заданию №1 пользователь вводит с клавиатуры последовательность, состоящей из 20 действительных чисел, и программа, представленная на Листинге 1, находит наименьшее из них [2].

```
Sub Zadanie1Sub()
Dim smallest As Double
Dim i As Integer
smallest = InputBox("Enter a number:")
For i = 1 To 19
    Dim num As Double
    num = InputBox("Enter a number:")
    If num < smallest Then
        smallest = num
    End If
Next i
MsgBox "The smallest number is " & smallest
End Sub
```

Листинг 1 – Задание №1

Согласно Листингу 1, вначале объявляются две переменные: "smallest" (наименьшее число) и "i" (счетчик цикла). Переменная "smallest" инициализируется значением, введенным пользователем с помощью функции InputBox, которая отображает диалоговое окно для ввода значения. Затем следует цикл "For", который выполняется 19 раз.

Внутри цикла каждый раз запрашивается у пользователя число, и это число сохраняется в переменной "num". Далее происходит проверка условия: если текущее число "num" меньше, чем значение переменной "smallest", то значение переменной "smallest" обновляется значением "num". Таким образом, после каждой итерации цикла, переменная "smallest" будет содержать наименьшее число из всех введенных. По завершении цикла, отображается сообщение MsgBox с результатом, которое сообщает пользователю наименьшее число, найденное во время выполнения программы.

Согласно заданию №2 при помощи датчика случайных чисел вводится последовательность $A(i)$, состоящая из N (случайное число от 7 до 30) целых положительных случайных трехзначных чисел. Нужно получить из последовательности $A(i)$ получить последовательность $B(i)$, переставив в элементах последовательности A первую и вторую цифру. Обе последовательности вывести в столбцы A и B рабочего листа. Данная программа реализована на Листинге 2.

```

Sub Task()
  Dim N As Integer
  N = Int((30 - 7 + 1) * Rnd + 7) 'генерируется случайное число от 7 до 30
  Dim A() As Integer
  ReDim A(1 To N)
  Dim i As Integer
  For i = 1 To N
    A(i) = Int((999 - 100 + 1) * Rnd + 100) 'генерируется случайное 3-х значное число
  Next i
  Dim B() As Integer
  ReDim B(1 To N)
  For i = 1 To N
    Dim hundreds As Integer
    Dim tens As Integer
    Dim units As Integer
    hundreds = A(i) \ 100
    tens = (A(i) Mod 100) \ 10
    units = A(i) Mod 10
    b(i) = tens * 100 + hundreds * 10 + units 'переставляются первая и вторая цифра 0
  Next i
  Dim ws As Worksheet
  Set ws = ThisWorkbook.ActiveSheet
  Dim j As Integer
  For j = 1 To N
    ws.Cells(j, 1).Value = A(j) 'выводим a на столбец A
    ws.Cells(j, 2).Value = B(j) 'выводим b на столбец B
  Next j
End Sub

```

Листинг 2 – Задание №2

Согласно заданию №2, вначале объявляется переменная «N» типа Integer, которая будет использоваться для хранения случайного числа от 7 до 30. Затем с помощью функции Rnd и математических операций генерируется случайное число в указанном диапазоне, которое присваивается переменной «N».

Объявляется массив «A» типа Integer, размерность которого задается равной числу «N». Этот массив будет использоваться для хранения случайных трехзначных чисел.

Выполняется цикл «For», в котором генерируются случайные трехзначные числа с помощью функции Rnd и математических операций. Сгенерированное число присваивается элементу массива «A» с текущим индексом «i».

Объявляется второй массив «B» типа Integer с той же размерностью «N». Выполняется второй цикл «For» для каждого элемента массива «A». Внутри цикла текущее трехзначное число из массива «A» разбивается на отдельные цифры с помощью операций деления и взятия остатка от деления. Затем цифры переставляются местами, чтобы первая и вторая цифра поменялись местами, а третья осталась на месте. Полученное переставленное число присваивается элементу массива «B» с текущим индексом «i».

Создается объект «ws» типа Worksheet, который представляет активный лист книги. Затем выполняется цикл «For» для каждого индекса «j» от 1 до «N». Внутри цикла значения элементов массивов «A» и «B» записываются в ячейки столбцов A и B соответственно на активном листе.

Согласно заданию №3 в одномерный массив «A» размерности N, вводимой с клавиатуры, при помощи датчика случайных чисел вводятся действительные числа в диапазоне от 900 до 4500. Их следует вывести в строку рабочего листа и выделить цветом все числа, в которых целая часть кратна 4. Данная программа реализована на Листинге 3.

```
Sub RandomNumbers()
    Dim N As Integer
    N = InputBox("Введите размерность массива:")
    Dim A() As Double
    ReDim A(1 To N)
    Dim I As Integer
    For I = 1 To N
        A(i) = Int((4500 - 900 + 1) * Rnd + 900) 'генерация случайных чисел в диапазоне от 900 до 4500
    Next I
    For I = 1 To N
        Cells(1, i) = A(i) 'вывод чисел в строку рабочего листа
        If Int(A(i)) Mod 4 = 0 Then 'если целая часть числа кратна 4
            Cells(1, i).Interior.Color = RGB(255, 0, 0) 'выделение цветом
        End If
    Next i
End Sub
```

Листинг 3 – Задание №3

Согласно заданию №3, вначале объявляется переменная «N» типа Integer, которая будет использоваться для хранения размерности массива. Затем объявляется массив «A» типа Double, размерность которого задается равной числу «N». Этот массив будет использоваться для хранения сгенерированных случайных чисел.

Выполняется цикл «For», в котором генерируются случайные числа в диапазоне от 900 до 4500 с помощью функции Rnd и математических операций. Сгенерированное число присваивается элементу массива «A» с текущим индексом «i».

Затем выполняется второй цикл «For» для каждого элемента массива «A». Внутри цикла проверяется условие: если целая часть текущего числа «A(i)» кратна 4, то выполняется следующая строка кода. Внутри условия текущая ячейка активного листа книги присваивается значению элемента массива A(i). Если условие истинно, то цвет фона ячейки изменяется на красный.

Согласно заданию №4 следует написать программу, представленную на Листинге 4, которая вычисляет функцию и строит график на отрезке [0;2], а также касательную к ней в точке с абсциссой $X_0 = 0.2[3]$.

```

With Лист1
.Range("A1").Value = "x"
.Range("B1").Value = "F(x)"
.Range("C1").Value = "Касательная"
End With
For x = 0 To 2 Step 0.01
    power = 1.2
    denominator = 2
    sum = 0
    Do While power <= 4
        sum = sum + Sin(x ^ power) / denominator
        power = power + 0.2
        denominator = denominator * 2
    Loop
    If x = 0.2 Then
        x0 = x
        tangent = sum
    End If
    With Лист1
        .Cells(.Cells(.Rows.Count, 1).End(xlUp).Row + 1, 1).Value = x
        .Cells(.Cells(.Rows.Count, 2).End(xlUp).Row + 1, 2).Value = sum
    End With
Next x
With Лист1.ChartObjects.Add(Left:=350, Width:=400, Top:=20, Height:=300)
    .Chart.SetSourceData Source:=Лист1.Range("A1:B" & Лист1.Cells(Лист1.Rows.Count,
1).End(xlUp).Row)
    .Chart.ChartType = xlXYScatterLines
    .Chart.HasTitle = True
    .Chart.ChartTitle.Text = "Граф F(x)"
    .Chart.Legend.Position = xlLegendPositionBottom
End With
With Лист1.ChartObjects(1).Chart.SeriesCollection.NewSeries
    .Name = "Tangent"
    .Values = Array(0, tangent)
    .XValues = Array(0, x0)
    .MarkerStyle = xlMarkerStyleNone
    .Format.Line.ForeColor.RGB = RGB(255, 0, 0) ' Красный цвет линии
    .Format.Line.Weight = 1.5 ' Толщина линии
End With

```

Листинг 4 – Задание №4

Согласно заданию №4, объявляются переменные «x», «result», «power», «denominator» и «sum» типа Double, которые будут использоваться для вычислений. Объявляются переменные «x0» и «tangent» типа Double, которые будут использоваться для хранения значения точки и значения касательной в этой точке соответственно.

Создается график на «Лист1». В ячейках A1, B1 и C1 записываются заголовки для столбцов «x», «F(x)» и «Касательная».

Выполняется цикл «For» для значения переменной «x» от 0 до 2 с шагом 0.01. Внутри цикла обнуляются переменные «power», «denominator» и «sum». Затем с помощью цикла «Do

While» вычисляется функция $F(x)$ путем суммирования значений функции $\text{Sin}(x^{\text{power}})/\text{denominator}$.

Переменные «power» и «denominator» изменяются в каждой итерации цикла. При $x = 0.2$ вычисленные значения $x0$ и «tangent» присваиваются соответствующим переменным. Записываются значения переменных «x» и «sum» в таблицу на «Лист1».

Создается график на «Лист1» с использованием функции `ChartObjects.Add`. Графику присваивается тип `xlXYScatterLines`. Создается новая серия данных «Tangent» для графика, которая представляет касательную. Задаются значения X и Y для серии, стиль линии.

Согласно заданию №5 нужно написать программу, представленную на Листинге 5 и Листинге 6, которая вычисляет функцию и автоматически строит график данной функции на произвольном отрезке [a;b] (a и b вводятся с клавиатуры).

```

Sub SolveAndPlotFunction()
    x = CDBl(InputBox("Введите значение x:"))
    x1 = CDBl(InputBox("Введите значение x1:"))
    Dim stepSize As Double
    stepSize = 0.1
    Dim lastRow As Long
    lastRow = ThisWorkbook.Sheets("Лист1").Cells(ThisWorkbook.Sheets("Лист1").Rows.Count, "A").End(xlUp).Row
    Dim dataRange As Range
    Set dataRange = ThisWorkbook.Sheets("Лист1").Range("A2:B" & (lastRow + (x1 - x) / stepSize))
    FillFunctionTable x, x1, stepSize, dataRange
    PlotFunctionChart dataRange
End Sub
Sub FillFunctionTable(ByVal x As Double, ByVal x1 As Double, ByVal stepSize As Double, ByRef dataRange As Range)
    dataRange.ClearContents
    Dim currentValue As Double
    currentValue = x
    Dim i As Long
    i = 1
    While currentValue <= x1
        dataRange.Cells(i, 1).Value = currentValue
        dataRange.Cells(i, 2).Value = FunctionValue(currentValue)
        currentValue = currentValue + stepSize
        i = i + 1
    Wend
End Sub

```

Листинг 5 – Построение значений функции

Функция `SolveAndPlotFunction`, представленная на Листинге 5, является основной процедурой, которая вызывает другие процедуры. Процедура `FillFunctionTable` принимает значения «x», «x1» и шага (`stepSize`) и заполняет таблицу значений функции в указанном диапазоне

dataRange. Она использует цикл While для перебора значений в диапазоне «x» до «x1» с указанным шагом.

```
Function FunctionValue(ByVal x As Double) As Double
    If x < -1 Then
        FunctionValue = 0
    ElseIf x >= -1 And x < 0 Then
        FunctionValue = Cos(WorksheetFunction.Pi() * x)
    ElseIf x >= 0 And x < 2 Then
        FunctionValue = x ^ 2 + 1 - x
    ElseIf x >= 2 And x < 7 Then
        FunctionValue = 7 - x
    Else
        FunctionValue = 0
    End If
End Function
Sub PlotFunctionChart(ByVal dataRange As Range)
    Dim chartObject As chartObject
    Set chartObject = ActiveSheet.ChartObjects.Add(Left:=10, Width:=400, Top:=10, Height:=300)
    Dim chart As chart
    Set chart = chartObject.chart
    With chart
        .ChartType = xlXYScatterLines
        .SeriesCollection.NewSeries
        .SeriesCollection(1).XValues = dataRange.Columns(1)
        .SeriesCollection(1).Values = dataRange.Columns(2)
        .HasTitle = True
        .ChartTitle.Text = "График функции"
        .Axes(xlCategory).HasTitle = True
        .Axes(xlCategory).AxisTitle.Text = "x"
        .Axes(xlValue).HasTitle = True
        .Axes(xlValue).AxisTitle.Text = "y"
    End With
End Sub
```

Листинг 6 – Построение графика функции

На каждой итерации процедура вычисляет значение функции с помощью функции FunctionValue, представленной на Листинге 6, и записывает соответствующие значения x и y в таблицу.

Согласно заданию №5, решается задача построения таблицы значений функции и графика функции на основе введенных пользователем значений «x» и «x1».

Функция FunctionValue принимает значение x и вычисляет соответствующее значение функции. Если x меньше -1, функция возвращает 0. Если «x» находится в диапазоне от -1 до 0, функция возвращает значение косинуса от «x», умноженного на π. Если «x» находится в диапазоне от 0 до 2, функция возвращает значение $x^2 + 1 - x$. Если «x» находится в диапазоне

от 2 до 7, функция возвращает значение 7 - x. Во всех остальных случаях функция возвращает 0.

Процедура PlotFunctionChart принимает диапазон данных dataRange и создает новый объект графика на активном листе. Затем она устанавливает необходимые свойства для графика. После выполнения кода будет отображена форма с запросом пользователю для ввода значений «x» и «x1». Затем программа заполнит таблицу значений функции и построит график на основе введенных значений.

Согласно заданию №6 в массив A считывается 100 действительных чисел, находящиеся в первом столбце рабочего листа. Следует отсортировать элементы, стоящие на нечетных местах по возрастанию, а на четных листах по убыванию элементов. Полученный массив записать во второй столбец и выделить разным цветом ячейки, соответствующие возрастающим и убывающим подмассивам. Данная программа реализована на Листинге 7 [4].

```

Set ws = ThisWorkbook.ActiveSheet
Set rngA = ws.Range("A1:A100")
arrA = rngA.Value
Set rngB = ws.Range("B1").Resize(UBound(arrA, 1), 1)
ascendingColor = RGB(255, 255, 0) ' Желтый цвет
descendingColor = RGB(255, 0, 0) ' Красный цвет
For i = 1 To UBound(arrA, 1) Step 2
    For j = i + 2 To UBound(arrA, 1) Step 2
        If arrA(j, 1) < arrA(i, 1) Then
            Swap arrA(i, 1), arrA(j, 1)
        End If
    Next j
    rngB.Cells(i, 1).Interior.Color = ascendingColor
Next i
For i = 2 To UBound(arrA, 1) Step 2
    For j = i + 2 To UBound(arrA, 1) Step 2
        If arrA(j, 1) > arrA(i, 1) Then
            Swap arrA(i, 1), arrA(j, 1)
        End If
    Next j
    rngB.Cells(i, 1).Interior.Color = descendingColor
Next i
rngB.Value = arrA
End Sub
Sub Swap(ByRef a As Variant, ByRef b As Variant)
    Dim temp As Variant
    temp = a
    a = b
    b = temp

```

Листинг 7 – Задание №6

Согласно заданию №6, в процедуре SortArray, переменная «ws» представляет активный лист, на котором выполняется сортировка. Переменная «rngA» указывает на столбец A, из

которого будут считаны числа для сортировки. Значения из столбца A считываются в массив «arrA».

Затем создается столбец B с помощью переменной «rngB», размер которого соответствует размеру массива «arrA». Сортировка массива выполняется с помощью двух вложенных циклов.

Внешний цикл перебирает нечетные позиции, а внутренний цикл выполняет сортировку нечетных позиций по возрастанию. После сортировки каждой нечетной позиции, ячейки в столбце B, соответствующие этим позициям, выделяются желтым цветом. Затем выполняется аналогичная сортировка для четных позиций, но уже по убыванию.

После сортировки каждой четной позиции, ячейки в столбце B, соответствующие этим позициям, выделяются красным цветом. Наконец, отсортированные значения записываются обратно в столбец B с помощью `rngB.Value = arrA`. Процедура `Swap` выполняет обмен значений между двумя переменными. В данном случае она используется для обмена значений в массиве «arrA» в процедуре `SortArray`.

Согласно заданию №7, при помощи датчика случайных чисел заполнить целочисленную квадратную матрицу A порядка «N» (N вводится с клавиатуры). Столбцы матрицы A, имеющие нечетную сумму элементов отсортировать по убыванию элементов. Затем следует вывести сумму элементов столбцов, а также матрицу A до и после сортировки. Все элементы отсортированных столбцов выделить цветом. Данная программа реализована на Листинге 8 и Листинге 9.

```
Sub SortColumnDescending(ByRef A() As Integer, ByVal N As Integer, ByVal columnIndex As Integer)
    Dim i As Integer, j As Integer
    For i = 1 To N - 1
        For j = i + 1 To N
            If A(i, columnIndex) < A(j, columnIndex) Then
                Swap A(i, columnIndex), A(j, columnIndex)
            End If
        Next j
    Next i
End Sub
Sub Swap(ByRef A As Integer, ByRef b As Integer)
    Dim temp As Integer
    temp = A
    A = b
    b = temp
End Sub
```

Листинг 8 – Сортировка матрицы

Процедура `SortColumnDescending` сортирует столбец по убыванию, данная процедура представлена на Листинге 8. Создается и заполняется массив `SortedA` копией матрицы A. Затем выполняется аналогичная сортировка столбцов матрицы `SortedA`.

```

Sub FillAndSortMatrix()
    N = InputBox("Введите порядок матрицы:")
    Set rng = Лист1.Range("A1").Resize(N, N)
    Dim randomGenerator As Object
    Set randomGenerator = CreateObject("System.Random")
    Dim i As Integer, j As Integer
    Dim A() As Integer
    ReDim A(1 To N, 1 To N)
    Dim columnSums() As Long
    ReDim columnSums(1 To N)
    Randomize
    For j = 1 To N
        Dim columnSum As Long
        columnSum = 0
        For i = 1 To N
            A(i, j) = Int((100 - (-100) + 1) * Rnd + (-100))
            columnSum = columnSum + A(i, j)
        Next i
        columnSums(j) = columnSum
    Next j
    rng.Value = A
    For j = 1 To N
        If columnSums(j) Mod 2 <> 0 Then
            SortColumnDescending A, N, j
        End If
    Next j
    Dim SortedA() As Integer
    ReDim SortedA(1 To N, 1 To N)
    For i = 1 To N
        For j = 1 To N
            SortedA(i, j) = A(i, j)
        Next j
    Next i
    For j = 1 To N
        If columnSums(j) Mod 2 <> 0 Then
            SortColumnDescending SortedA, N, j
        End If
    Next j
    rng.Offset(N + 2).Value = SortedA
    For j = 1 To N
        If columnSums(j) Mod 2 <> 0 Then
            Dim sortedColumnRange As Range
            Set sortedColumnRange = rng.Offset(N + 2).Columns(j)
            sortedColumnRange.Interior.Color = RGB(211, 254, 171) ' Измените цвет на нужный вам
        End If
    Next j
    Dim sumRange As Range
    Set sumRange = rng.Resize(, 1).Offset(, N + 1)
    sumRange.Value = Application.WorksheetFunction.Transpose(columnSums)
End Sub

```

Листинг 9 – Построение матрицы

Согласно заданию №7, код на языке VBA выполняет заполнение и сортировку матрицы на листе "Лист1". Пользователю предлагается ввести порядок матрицы. Затем матрица

заполняется случайными числами в диапазоне от -100 до 100. После заполнения матрицы происходит сортировка столбцов в зависимости от суммы элементов каждого столбца.

Процедура FillAndSortMatrix, представленная на Листинге 9, начинается с запроса пользователя на ввод порядка матрицы. Затем создается диапазон «rng», указывающий на ячейку "A1" и растягивающийся на «N» строк и «N» столбцов. Массив «A» размером «N» x «N» и массив columnSums размером «N» используются для хранения матрицы и сумм элементов каждого столбца соответственно.

Затем используется объект System.Random для генерации случайных чисел. С помощью вложенных циклов матрица «A» заполняется случайными числами, а также вычисляется сумма элементов каждого столбца, которая записывается в массив «columnSums».

Далее неотсортированная матрица «A» выводится в диапазон «rng». Затем происходит сортировка столбцов матрицы «A» в соответствии с суммами элементов каждого столбца. Если сумма элементов столбца является нечетной, вызывается процедура SortColumnDescending.

Отсортированная матрица SortedA выводится в диапазон «rng» с смещением на N + 2 строк. Цветом выделяются столбцы, которые были отсортированы. Если сумма элементов столбца является нечетной, то соответствующий столбец в диапазоне «rng.Offset». Columns(j) выделяется цветом. Наконец, сумма элементов каждого столбца выводится в столбец справа от матрицы, в диапазон «sumRange».

Согласно заданию №8, во введенном с клавиатуры предложении нужно поменять порядок слов на обратный. Программа представлена на Листинге 10 [5].

```
inputSentence = InputBox("Введите предложение:")
words = Split(inputSentence, " ")
For i = UBound(words) To LBound(words) Step -1
    reversedSentence = reversedSentence & " " & words(i)
Next i
reversedSentence = Trim(reversedSentence)
reversedSentence = Replace(reversedSentence, ",,", ", ")
MsgBox "Результат: " & reversedSentence
```

Листинг 10 – Задание №8

Согласно заданию №8, процедура ReverseSentence начинается с запроса пользователя на ввод предложения. Введенное предложение сохраняется в переменную «inputSentence». Далее, предложение разбивается на отдельные слова с помощью функции Split. Разделителем является пробел. Результат сохраняется в массив строк «words». Затем происходит переворачивание порядка слов в предложении. С помощью цикла «For» перебираются индексы массива «words» в обратном порядке. Каждое слово добавляется в переменную «reversedSentence» с пробелом перед ним. После завершения цикла, начальный пробел удаляется с помощью

функции Trim. Также учитывается пунктуация, и к запятым добавляется пробел с помощью функции Replace. Наконец, результат выводится в сообщении с помощью функции MsgBox, которая содержит текст "Результат: " и переменную «reversedSentence».

Согласно заданию №9, нужно написать логическую функцию, принимающая значения True, если в целом числе, являющимся единственным аргументом этой функции, сумма цифр – простое число. Данная программа реализована на Листинге 11.

```
Function IsDigitSumPrime(ByVal number As Long) As Boolean
```

```

    Dim digitSum As Long
    Dim i As Long
    digitSum = 0
    Do While number <> 0
        digitSum = digitSum + (number Mod 10)
        number = number \ 10
    Loop
    If digitSum < 2 Then
        IsDigitSumPrime = False
    Else
        For i = 2 To Sqr(digitSum)
            If digitSum Mod i = 0 Then
                IsDigitSumPrime = False
                Exit Function
            End If
        Next i
        IsDigitSumPrime = True
    End If
End Function
Sub WriteResultToTable()
    Dim inputRange As Range
    Dim cell As Range
    Set inputRange = Range("A1:A10")
    For Each cell In inputRange
        cell.Offset(0, 1).Value = IsDigitSumPrime(cell.Value)
    Next cell
    MsgBox "Результаты в таблице"
End Sub
```

Листинг 11 – Задание №9

Согласно заданию №9, функция IsDigitSumPrime принимает один параметр «number» типа Long и возвращает значение типа Boolean. Эта функция проверяет, является ли сумма цифр числа простым числом.

Сначала в функции вычисляется сумма цифр числа «number». Для этого используется цикл Do While. В каждой итерации цикла остаток от деления числа на 10 добавляется к переменной «digitSum», а затем число делится нацело на 10. Цикл продолжается, пока число не равно 0.

Затем функция проверяет, является ли сумма цифр «digitSum» менее 2. Если это так, функция возвращает False. Если сумма цифр «digitSum» больше или равна 2, функция выполняет цикл «For» для проверки, является ли сумма цифр простым числом.

Цикл начинается с числа 2 и продолжается до квадратного корня из «digitSum». В каждой итерации цикла проверяется, делится ли «digitSum» на текущее значение «i» без остатка. Если деление без остатка выполняется, то сумма цифр не является простым числом, и функция возвращает False. Если цикл завершается без нахождения делителей, то сумма цифр является простым числом, и функция возвращает True.

Процедура WriteResultToTable используется для записи результатов функции IsDigitSumPrime в таблицу. Сначала указывается диапазон ячеек «inputRange», в котором содержатся целые числа. Затем происходит цикл «For Each», который перебирает каждую ячейку в «inputRange».

Для каждой ячейки вызывается функция IsDigitSumPrime, и результат записывается в соседнюю ячейку с помощью cell.Offset(0, 1).Value. По завершении цикла выводится сообщение в виде диалогового окна с текстом "Результаты в таблице" с помощью функции MsgBox.

Согласно заданию №10, в первых 10 столбцах рабочего листа находятся сведения о сотрудниках фирмы. При этом в первых трех столбцах рабочего листа записаны: фамилия, имя, отчество. Пользователь самостоятельно заполняет и записывает информацию во все столбцы и поля. Необходимо реализовать программу, представленную на Листинге 12 и 13, считывающую все данные и сортирующую этот список по возрастанию количества букв, содержащих в первых трех столбцах. Результат выводится на другой рабочий лист [6].

```

Application.ScreenUpdating = False
Sheets("Лист2").Range("A1:D12").Clear
With Sheets("Лист1")
    .Select
    iLastRow = .Cells(.Rows.Count, 1).End(xlUp).row
    Arr() = .Range("A1").CurrentRegion.Value
    ReDim Preserve Arr(1 To UBound(Arr, 1), 1 To UBound(Arr, 2) + 1)
    For i = LBound(Arr, 2) To UBound(Arr, 2)
        Arr(i, UBound(Arr, 2)) = Len(Arr(i, 1)) + Len(Arr(i, 2)) + Len(Arr(i, 2))
    Next i
    Arr() = CoolSort(Arr(), UBound(Arr, 2))
End With
With Sheets("Лист2")
    .Select
    .Range("A1").Resize(UBound(Arr, 1), UBound(Arr, 2) - 1).Value = Arr()
End With

```

Листинг 12 – Подготовка к сортировке

Согласно заданию №10, в процедуре SortEmployees, представленной на Листинге 12, происходит подготовка к сортировке. Сначала очищается диапазон A1:D12 на "Лист2". Затем

данные считываются с "Листа1" в массив «Arr» с помощью метода CurrentRegion, который выбирает связанный диапазон данных, начиная с ячейки A1. Информация о массиве «Arr» хранится в формате вариантов данных.

```
Function CoolSort(SourceArr As Variant, ByVal N As Integer) As Variant
  If N > UBound(SourceArr, 2) Or N < LBound(SourceArr, 2) Then _
    MsgBox "Нет столбца", vbCritical: Exit Function
  Dim Check As Boolean, iCount As Integer, jCount As Integer
  ReDim tmpArr(UBound(SourceArr, 2)) As Variant
  Do Until Check
    Check = True
    For iCount = LBound(SourceArr, 1) To UBound(SourceArr, 1) - 1
      If Val(SourceArr(iCount, N)) > Val(SourceArr(iCount + 1, N)) Then
        For jCount = LBound(SourceArr, 2) To UBound(SourceArr, 2)
          tmpArr(jCount) = SourceArr(iCount, jCount)
          SourceArr(iCount, jCount) = SourceArr(iCount + 1, jCount)
          SourceArr(iCount + 1, jCount) = tmpArr(jCount)
        Next
        Check = False
      Next
    End If
  Next
  CoolSort = SourceArr
End Function
```

Листинг 13 – Сортировка массива

Размер массива «Arr» изменяется, добавляя один столбец в конец с помощью оператора ReDim Preserve. В этом столбце будет записана сумма длин значений из первых трех столбцов каждой строки.

Происходит цикл For для каждого столбца в «Arr». Значение суммы длин трех первых столбцов каждой строки вычисляется с помощью функции Len и записывается в последний столбец массива «Arr».

Вызывается пользовательская функция CoolSort, которая принимает массив «Arr» и номер столбца N для сортировки.

В функции CoolSort, представленной на Листинге 13, происходит сортировка массива «SourceArr» по столбцу с номером N. Используется алгоритм сортировки пузырьком. Цикл «Do Until» выполняется, пока не будет выполнено условие Check = True, которое означает, что массив отсортирован.

Внутри цикла происходит сравнение значений в столбце «N» двух соседних строк массива «SourceArr». Если значение текущей строки больше значения следующей строки, то значения меняются местами с помощью временного массива «tmpArr», и флаг Check устанавливается в False для указания, что была произведена перестановка.

После завершения сортировки, отсортированные данные записываются из массива «SourceArr» на "Лист2". В результате выполнения процедуры SortEmployees, данные на "Листе2" будут отсортированы по указанному столбцу.

Согласно заданию №11, при помощи датчика случайных чисел нужно получить и вывести в столбец рабочего листа N целых чисел типа Long. В соседний столбец вывести представление соответствующего числа в дополнительном коде шестнадцатеричной системы счисления. Данная программа реализована на Листинге 14.

```
Sub GenerateRandomNumbers()
    Dim N As Integer
    N = InputBox("Введите количество чисел N:")
    Dim sheet As Worksheet
    Set sheet = ThisWorkbook.Sheets("Лист1")
    Randomize
    Dim randomRange As Range
    Set randomRange = sheet.Range("A1:A" & N)
    Dim hexRange As Range
    Set hexRange = sheet.Range("B1:B" & N)
    Dim i As Integer
    For i = 1 To N
        Dim randomNumber As Long
        randomNumber = RandomNumberInRange(0, 999999999)
        randomRange.Cells(i).Value = randomNumber
        hexRange.Cells(i).Value = Hex(randomNumber)
    Next i
End Sub
Function RandomNumberInRange(ByVal min As Long, ByVal max As Long) As Long
    RandomNumberInRange = CLng(((max - min + 1) * Rnd) + min)
    Do While RandomNumberInRange > max
        RandomNumberInRange = CLng(((max - min + 1) * Rnd) + min)
    Loop
End Function
```

Листинг 14 – Задание №11

Согласно заданию №11, в процедуре GenerateRandomNumbers запрашивается у пользователя количество чисел «N». Определяется рабочий лист с помощью переменной «sheet». Вызывается функция Randomize для инициализации генератора случайных чисел. Создаются объекты диапазонов «randomRange» и «hexRange», которые представляют столбцы A и B соответственно на рабочем листе. Размер диапазонов зависит от значения N. В цикле For от 1 до «N» генерируются случайные числа с помощью вызова функции RandomNumberInRange. Сгенерированное случайное число выводится в столбец A с помощью свойства Value объекта ячейки внутри диапазона «randomRange». Шестнадцатеричное представление числа выводится в соседний столбец B с помощью функции Hex.

Функция RandomNumberInRange принимает минимальное и максимальное значения и возвращает случайное число в заданном диапазоне. Внутри функции используется функция

Rnd для генерации случайного числа от 0 до 1. Если случайное число превышает максимальное значение, генерация повторяется до тех пор, пока не будет получено число в допустимом диапазоне.

Согласно заданию №12, нужно написать программу, представленную на Листингах 15,16 и 17, которая умножает сверхбольшие целые числа, и следует определить сумму цифр в сверхбольшом числе. Далее построить гистограмму частоты появления цифр в полученном сверхбольшом числе.

```
Function MultiplyLargeNumbers(numberA As String, numberB As String) As String
    lengthA = Len(numberA)
    lengthB = Len(numberB)
    lengthResult = lengthA + lengthB
    result = String(lengthResult, "0")
    For i = lengthA To 1 Step -1
        digitA = CInt(Mid(numberA, i, 1))
        carry = 0
        For j = lengthB To 1 Step -1
            digitB = CInt(Mid(numberB, j, 1))
            product = digitA * digitB + carry + CInt(Mid(result, i + j, 1))
            carry = product \ 10
            sum = product Mod 10
            Mid(result, i + j, 1) = CStr(sum)
        Next j
        If carry > 0 Then
            Mid(result, i + j, 1) = CStr(carry)
        End If
    Next i
    MultiplyLargeNumbers = result
End Function
```

Листинг 15 – Умножение сверхбольших чисел

Вызывается функция MultiplyLargeNumbers, представленной в Листинге 15, для умножения сверхбольших чисел «numberA» и «numberB». Результат сохраняется в переменной «result».

```
Function SumDigits(number As String) As Integer
    Dim sum As Integer
    Dim i As Integer
    sum = 0
    For i = 1 To Len(number)
        sum = sum + CInt(Mid(number, i, 1))
    Next i
    SumDigits = sum
End Function
```

Листинг 16 – Вычисления суммы цифр

Вызывается функция SumDigits, представленная на Листинге 16, для вычисления суммы цифр в полученном результате «result». Сумма сохраняется в переменной «digitSum». Создается объект «digitFrequency» класса Scripting.Dictionary для хранения частоты появления цифр в «result».

```

Sub MultiplyAndHistogram()
    numberA = "1234567890123456789012345678901234567890"
    numberB = "9876543210987654321098765432109876543210"
    result = MultiplyLargeNumbers(numberA, numberB)
    digitSum = SumDigits(result)
    Set digitFrequency = CreateObject("Scripting.Dictionary")
    For i = 1 To Len(result)
        digit = Mid(result, i, 1)
        If digitFrequency.Exists(digit) Then
            digitFrequency(digit) = digitFrequency(digit) + 1
        Else
            digitFrequency.Add digit, 1
        End If
    Next i
    MsgBox "Сумма цифр в числе: " & digitSum
    Dim chartRange As Range
    Set chartRange = ThisWorkbook.Worksheets(1).Range("A1:B11")
    chartRange.ClearContents
    chartRange.Cells(1, 1).Value = "Цифра"
    chartRange.Cells(1, 2).Value = "Частота"
    i = 2
    For Each digit In digitFrequency.Keys
        chartRange.Cells(i, 1).Value = digit
        chartRange.Cells(i, 2).Value = digitFrequency(digit)
        i = i + 1
    Next digit
    Dim chartObject As chartObject
    Set chartObject = ThisWorkbook.Worksheets(1).ChartObjects.Add(Left:=200, Top:=200, Width:=400,
Height:=300)
    With chartObject.Chart
        .SetSourceData chartRange
        .ChartType = xlColumnClustered
        .HasTitle = True
        .ChartTitle.Text = "Гистограмма частоты появления цифр"
        .Axes(xlCategory).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Text = "Цифры"
        .Axes(xlValue).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Text = "Частота"
    End With
End Sub

```

Листинг 17 – Построение гистограммы и вычисления суммы цифр

В процедуре MultiplyAndHistogram, представленной в Листинге 17, задаются значения двух сверхбольших чисел «numberA» и «numberB».

В цикле проходятся все цифры в «result». Если цифра уже присутствует в словаре digitFrequency, увеличивается соответствующее ей значение на 1. Если цифра отсутствует в

словаре, она добавляется со значением 1. Выводится сообщение с суммой цифр в числе «digitSum».

Создается диапазон «chartRange» на первом листе, в который будут записываться значения для гистограммы.

Записываются заголовки столбцов "Цифра" и "Частота" в «chartRange». В цикле «For Each» проходятся все ключи в словаре digitFrequency. Значения цифр и их частоты записываются в диапазон «chartRange».

Создается объект гистограммы «chartObject» на первом листе с заданными размерами и расположением.

Функция MultiplyLargeNumbers принимает две строки «numberA» и «numberB» и выполняет умножение сверхбольших чисел. Внутри функции происходит алгоритмическое умножение с использованием столбикового метода. Результат умножения возвращается как строка «result». Функция SumDigits принимает строку «number» и вычисляет сумму всех цифр в этой строке. Сумма возвращается как целое число.

Согласно заданию №13, следует считать с первого столбца рабочего листа «N», вводимой с клавиатуры, вещественных чисел типа Singel. Перевести эти числа в шестнадцатеричную систему счисления и вывести в соседний столбец. В шестнадцатеричном представлении числа поменять 1-ую цифру с 8-ой, 2-ую с 7-ой и т.д. Полученное число перевести в десятичную систему счисления и вывести в 3 столбец рабочего листа. Данная программа реализована на Листингах 18 и 19 [7].

```
Function SwapHexDigits(hexValue As String) As String
    length = Len(hexValue)
    swappedHex = ""
    For i = 1 To length
        swappedHex = Mid(hexValue, i, 1) & swappedHex
    Next i
    SwapHexDigits = swappedHex
End Function
```

Листинг 18 – Изменение местами числа в шестнадцатеричном формате

Функция SwapHexDigits, представленная на Листинге 18, меняет местами цифры в шестнадцатеричном представлении числа «hexValue». Полученное новое шестнадцатеричное представление сохраняется в переменную «swappedHex». Для этого в начале кода объявляются переменные «length» и «swapped Hex». Переменная length устанавливается равной длине входного значения «hexValue». Затем идет цикл For, который выполняется от 1 до «length». С помощью функции Mid извлекается один символ из «hexValue» с помощью индекса i, и этот символ добавляется перед текущим значением «swappedHex». По мере итераций цикла символы извлекаются справа налево и добавляются в «swappedHex» в обратном порядке.

```

Sub ConvertAndSwapHex()
    N = InputBox("Введите количество чисел для обработки:")
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets(1)
    For i = 1 To N
        value = ws.Cells(i, 1).value
        hexValue = ""
        quotient = value
        Do Until quotient = 0
            remainder = quotient Mod 16
            If remainder < 10 Then
                hexValue = Chr(48 + remainder) & hexValue
            Else
                hexValue = Chr(55 + remainder) & hexValue
            End If
            quotient = Int(quotient / 16)
        Loop
        swappedHex = SwapHexDigits(hexValue)
        decimalValue = 0
        For k = 1 To Len(hexValue)
            Select Case Mid(hexValue, k, 1)
                Case "0" To "9"
                    digitValue = CInt(Mid(hexValue, k, 1))
                Case "A", "a"
                    digitValue = 10
                Case "B", "b"
                    digitValue = 11
                Case "C", "c"
                    digitValue = 12
                Case "D", "d"
                    digitValue = 13
                Case "E", "e"
                    digitValue = 14
                Case "F", "f"
                    digitValue = 15
            End Select
            decimalValue = decimalValue + digitValue * 16 ^ (Len(hexValue) - k)
        Next k
        ws.Cells(i, 2).value = swappedHex
        ws.Cells(i, 3).value = decimalValue
    Next i
End Sub

```

Листинг 19 – Перевод в шестнадцатеричный формат и обратно в десятичный формат

Согласно заданию №13, в процедуре ConvertAndSwapHex, представленной на Листинге 19, считывается значение «N» с клавиатуры, которое указывает количество чисел, которые нужно обработать. Определяется рабочий лист «ws».

В цикле проходятся все числа от 1 до N и выполняются следующие действия:

- значение числа считывается из первого столбца на текущей строке и сохраняется в переменную «value».

- происходит перевод числа в шестнадцатеричный формат. Полученное шестнадцатеричное представление сохраняется в переменную «hexValue».

- выполняется функция SwapHexDigits.

- происходит перевод шестнадцатеричного значения обратно в десятичную систему.

Результат сохраняется в переменную «decimalValue».

- значения «swappedHex» и «decimalValue» выводятся во второй и третий столбцы соответственно на текущей строке.

Согласно заданию №14, нужно написать программу, представленную на Листинге 20 и Листинге 21, которая считывает в одномерный массив А целые числа, записанные в столбец А рабочего листа, начиная с выделенной ячейки. Ячейки, в которых находятся числа, в разложении которых на простые множители, получается наибольшее число множителей, выделить цветом, а в соседний столбец вывести разложение таких чисел на множители.

```
Function Factorize(ByVal number As Long) As String
    factors = ""
    factor = 2
    While factor <= number
        If number Mod factor = 0 Then
            factors = factors & factor & " "
            number = number \ factor
        Else
            factor = factor + 1
        End If
    Wend
    Factorize = factors
End Function
```

Листинг 20 – Разложение число на простые множители

Функция Factorize, представленная на Листинге 20, разлагает число «currentNumber» на простые множители и возвращает результат в виде строки, которая записывается в соседний столбец В на текущей строке.

Объявляются переменные «factors» и «factor». Затем следует цикл While, который выполняется, пока «factor» меньше или равен «number». Внутри цикла проверяется, делится ли «number» на «factor» без остатка с помощью оператора модуля %. Если деление производится без остатка, то это означает, что «factor» является множителем «number». В таком случае, «factor» добавляется в строку «factors» с помощью оператора конкатенации &, а затем «number» делится на «factor» с использованием оператора целочисленного деления \, чтобы продолжить поиск других множителей числа «number». Если деление number на factor дает остаток, то это означает, что текущий «factor» не является множителем. В этом случае, значение factor увеличивается на 1.

```

Sub HighlightAndFactorizeNumbers()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Лист1")
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.count, 1).End(xlUp).Row
    Dim numbers() As Variant
    numbers = ws.Range("A1:A" & lastRow).Value
    Dim i As Long
    Dim maxFactorCount As Long
    maxFactorCount = 0
    For i = 1 To UBound(numbers, 1)
        Dim currentNumber As Long
        currentNumber = numbers(i, 1)
        Dim factorCount As Long
        factorCount = CountPrimeFactors(currentNumber)
        If factorCount > maxFactorCount Then
            maxFactorCount = factorCount
        End If
    Next i
    For i = 1 To UBound(numbers, 1)
        currentNumber = numbers(i, 1)
        factorCount = CountPrimeFactors(currentNumber)
        If factorCount = maxFactorCount Then
            ws.Cells(i, 1).Interior.Color = RGB(255, 189, 136)
        End If
        ws.Cells(i, 2).Value = Factorize(currentNumber)
    Next i
End Sub

Function CountPrimeFactors(ByVal number As Long) As Long
    Dim count As Long
    Dim factor As Long
    count = 0
    factor = 2
    While factor <= number
        If number Mod factor = 0 Then
            count = count + 1
            number = number \ factor
        Else
            factor = factor + 1
        End If
    Wend
    CountPrimeFactors = count
End Function

```

Листинг 21 – Выделение и нахождение наибольшего числа с множителями

Согласно заданию №14, в процедуре HighlightAndFactorizeNumbers, представленной на Листинге 21, определяется рабочий лист «ws». Находится последняя заполненная строка в столбце А с помощью функции «xlUp». Числа из столбца А записываются в массив «numbers». Создается переменная «maxFactorCount», которая будет содержать максимальное количество множителей для чисел.

В цикле проходятся все числа из массива «numbers», и выполняются следующие действия:

- значение текущего числа извлекается из массива «numbers» и сохраняется в переменную «currentNumber».
- вызывается функция CountPrimeFactors, которая подсчитывает количество простых множителей для числа «currentNumber» и возвращает результат в переменную «factorCount».
- если «factorCount» больше, чем текущее значение «maxFactorCount», то «maxFactorCount» обновляется.

Во втором цикле проходятся все числа из массива «numbers», и выполняются следующие действия:

- значение текущего числа извлекается из массива «numbers» и сохраняется в переменную «currentNumber».
- вызывается функция CountPrimeFactors, которая подсчитывает количество простых множителей для числа «currentNumber» и возвращает результат в переменную «factorCount».
- если «factorCount» равно «maxFactorCount», то ячейка в столбце A на текущей строке окрашивается в светло-оранжевый цвет.
- вызывается функция Factorize.

Согласно заданию №15, следует разработать игру «Морской бой» на языке VBA. Данная программа реализована на Листингах 22, 23, 24 [8].

```

Const BOARD_SIZE As Integer = 10
Const SHIP_COUNT As Integer = 5
Dim Board(1 To BOARD_SIZE, 1 To BOARD_SIZE) As String
Sub PlayBattleship()
    ClearBoard
    PlaceShips
    Do Until ShipsSunk = SHIP_COUNT
        DisplayBoard
        Shoot
        ShipsSunk = 0
        For i = 1 To SHIP_COUNT
            If IsShipSunk(i) Then
                ShipsSunk = ShipsSunk + 1
            End If
        Next i
    Loop
    DisplayBoard
    MsgBox "Поздравляшки! Теперь иди кушай!"
End Sub

```

Листинг 22 – Основная процедура игры

Процедура PlayBattleship, представленная на Листинге 22, является основной процедурой игры. Игра продолжается, пока все корабли не будут потоплены. По окончании выводится сообщение о победе.

```

Sub ClearBoard()

    For i = 1 To BOARD_SIZE
        For j = 1 To BOARD_SIZE
            Board(i, j) = "-"
        Next j
    Next i
End Sub

Sub DisplayBoard()
    For i = 1 To BOARD_SIZE
        For j = 1 To BOARD_SIZE
            boardString = boardString & Board(i, j) & " "
        Next j
        boardString = boardString & vbCrLf
    Next i
    MsgBox boardString
End Sub

Sub PlaceShips()
    For i = 1 To SHIP_COUNT
        PlaceShip i
    Next i
End Sub

Sub PlaceShip(shipNumber As Integer)
    Do
        row = Int((BOARD_SIZE * Rnd) + 1)
        col = Int((BOARD_SIZE * Rnd) + 1)
        Loop Until Board(row, col) = "-"
        Board(row, col) = "S" & shipNumber
    End Sub

Sub Shoot()
    Dim row As Integer, col As Integer
    Dim target As String
    row = InputBox("Введите номер строки (1-10):")
    col = InputBox("Введите номер столбца (1-10):")
    If Board(row, col) = "X" Or Board(row, col) = "*" Then
        MsgBox "Вы уже стреляли по этим координатам. Попробуйте снова."
        Shoot
        Exit Sub
    End If
    target = Board(row, col)
    If Left(target, 1) = "S" Then
        Board(row, col) = "X"
        MsgBox "ПопалЪ"
    Else
        Board(row, col) = "*"
        MsgBox "ПромахЪ"
    End If

End Sub

```

Листинг 23 – Очистение игровой доски, состояние доски, размещение кораблей, обстрел кораблей

Процедура ClearBoard, представленная на Листинге 23, очищает игровую доску, заполняя все ячейки символом "-".

Процедура DisplayBoard, представленная на Листинге 23, создает строку «boardString», содержащую текущее состояние игровой доски, и выводит эту строку в сообщении.

Процедура PlaceShips, представленная на Листинге 23, размещает корабли на игровой доске. Она вызывает процедуру PlaceShip для каждого корабля.

Процедура PlaceShip, представленная на Листинге 23, размещает один корабль на случайных координатах на игровой доске. Она выбирает случайные значения для «row» и «col» до тех пор, пока не найдет пустую ячейку на доске. Затем она помечает эту ячейку символом "S" и номером корабля.

Процедура Shoot, представленная на Листинге 23, позволяет игроку сделать выстрел. Она запрашивает у игрока координаты выстрела, проверяет, не стрелял ли игрок уже по этим координатам, а затем проверяет, попал ли игрок в корабль. Если попал, то ячейка на доске помечается символом "X", если промазал, то ячейка помечается символом "*".

```
Function IsShipSunk(shipNumber As Integer) As Boolean
    Dim i As Integer, j As Integer
    IsShipSunk = True
    For i = 1 To BOARD_SIZE
        For j = 1 To BOARD_SIZE
            If Board(i, j) = "S" & shipNumber Then
                IsShipSunk = False
                Exit Function
            End If
        Next j
    Next i
End Function
```

Листинг 24 – Проверка потопление корабля

Функция IsShipSunk, представленная на Листинге 24, проверяет, потоплен ли корабль с заданным номером. Она проходит по всем ячейкам игровой доски и проверяет, есть ли ячейка с символом "S" и номером корабля. Если такая ячейка найдена, то функция возвращает значение False, что означает, что корабль не потоплен. Если такой ячейки нет, то функция возвращает значение True, что означает, что корабль потоплен.

2.2 Инспектирование программного модуля на предмет соответствия стандартам кодирования

При написании кода на VBA следует придерживаться данных стандартов:

1) использовать логичную структура кода, включая определение переменных, объявление функций и констант.

2) использовать осмысленные названия для переменных, функций, констант и модулей.

3) использовать комментарии, чтобы объяснить сложные или неочевидные участки кода. Комментарии должны быть информативными и сжатыми, описывая цель и логику кода.

4) использовать правильное форматирование кода для облегчения его чтения. Отделять блоки кода пробелами и использованием отступов. Разделять операторы и аргументы пробелами для лучшей читабельности.

5) использовать строгую типизацию переменных. Объявлять переменные с использованием конкретных типов данных, чтобы избежать ошибок и неоднозначности.

6) избегать использования глобальных переменных во избежание конфликтов и неожиданного поведения программы. Использовать локальные переменные или передавать значения через параметры функций и процедур.

7) проверять ошибки и исключения в коде. Использовать конструкции обработки ошибок, такие как блоки Try-Catch, для предотвращения сбоев программы и улучшения ее надежности.

8) оптимизировать код, избегая излишнего использования циклов, рекурсивных вызовов или других медленных операций. Использовать эффективные алгоритмы и структуры данных.

9) использовать функции и процедуры для общих операций, которые могут быть использованы повторно в разных частях кода.

10) избегать использования магических чисел или строк в коде. Вместо этого объявлять константы или использовать переменные с понятными именами, чтобы облегчить понимание и изменение значений.

11) проверять вводимые данные на корректность и соблюдение формата.

12) разбивать длинные строки кода на несколько строк для лучшей читабельности. Использовать символ подчеркивания "_" для обозначения продолжения строки.

13) проводить регулярное тестирование кода для выявления ошибок и проблем. Использовать отладчик и тестовые данные для проверки правильности работы программы.

14) документировать код, включая описание функций, процедур и классов, а также спецификацию входных и выходных данных.

Собраны материалы практики, оформлен отчет и необходимые документы по практике согласно стандартам оформления документов такого рода.

Заключение

В ходе прохождения учебной практики был изучен материал, необходимый для написания отчета.

В процессе прохождения практики я получила навыки решения следующих задач по профессиональному модулю ПМ.01 «Осуществление интеграции программных модулей»:

- умение анализировать документацию и предложенные задания.
- умение разрабатывать план мероприятий для этапов разработки, тестирования программного модуля.

- умение разрабатывать программный модуль:

- 1) умение анализировать требования к программным модулям по предложенной документации.

- 2) умение разрабатывать алгоритм и структуру данных.

- 3) умение шлифовать текст модуля.

- 4) умение проверять модуль.

- 5) умение транслировать модуль.

- умение разрабатывать тестовые наборы и сценарии программного средства.

- умение инспектировать разработанные программные модули на предмет соответствия стандартам кодирования:

- 1) умение инспектировать функциональное тестирование текстовых наборов и сценариев программных средств.

- 2) умение инспектировать тестирование интеграции тестовых наборов и сценариев программных средств.

По окончании практики была достигнута цель - повышение профессиональной компетенции по профессиональному модулю ПМ.01 «Осуществление интеграции программных модулей». А также приобретены навыки и опыт практической работы.

Данная практика является хорошим практическим опытом для дальнейшей самостоятельной деятельности.

Список используемых источников

1. Skillfactory Media [сайт]. – URL: <https://blog.skillfactory.ru/> (дата обращения: 16.05.2023)
2. Н. Павлов. П12 Excel – готовые решения. Бери и пользуйся! / Николай Павлов. – Москва.: Книга по Требованию, 2018. – 382 с. – Текст: электронный. – URL: <https://excel16.files.wordpress.com/2016/03/microsoft-excel-d0b3d0bed182d0bed0b2d18bd0b5d180d0b5d188d0b5-d0bdd0b8d18f-d0b1d0b5.pdf> (дата обращения: 18.05.2023)
3. Кудрявцев Н.Г. Программирование на VBA MS Excel: учебное пособие / Н. Г. Кудрявцев, Д. В. Кудин, М. Ю. Беликова – Горно-Алтайск: РИО ГАГУ, 2018. – 116 с. – Текст: электронный. – URL: http://www.robionika.ru/uploads/files/book_vba.pdf (дата обращения: 21.05.2023)
4. И.С. Осетрова. Microsoft Visual Basic for Application: учебное пособие / И.С. Осетрова, Н. А. Осипов – СПб: НИУ ИТМО, 2018. – 120 с. – Текст электронный. – URL: <https://books.ifmo.ru/file/pdf/1088.pdf> (дата обращения: 23.05.2023)
5. Лебедев, В. М. Программирование на VBA в MS Excel: учебное пособие для среднего профессионального образования / В. М. Лебедев. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2023. – 306 с. – Текст: электронный. – URL: <https://urait.ru/bcode/513143> (дата обращения: 25.05.2023)
6. Р.Ш. Гайнанова. Программирование на Visual Basic for Applications в Excel: учебное пособие / Р. Ш. Гайнанова, О. А. Широкова – Казань: КФУ, 2019. – 153с. – Текст: электронный. URL: https://kpfu.ru/docs/F386355132/VBA_Excel.pdf (дата обращения: 26.05.2023)
7. Комолова, Н. В. Программирование на VBA в Excel 2019. Самоучитель / Н. В. Комолова, А. В. Клименко. – Санкт-Петербург: БХВ-Петербург, 2020. – 496 с. – Текст: электронный. – URL: https://balka-book.com/files/2021/06_07/11_37/u_files_store_25_18699.pdf (дата обращения: 28.05.2023)
8. Козлов, А. Ю. Статистический анализ данных в MS Excel: учебное пособие / А.Ю. Козлов, В.С. Мхитарян, В.Ф. Шишов. – Москва: ИНФРА-М, 2023. – 320 с. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1907518> (дата обращения: 30.05.2023)
9. Воробьева, В. Е. Основы численных методов и их реализация в MS Excel: учебное пособие / В. Е. Воробьева, Ф. И. Воробьева ; Минобрнауки России, Казан. нац. исслед. технол. ун-т. – Казань, 2022 – Текст: электронный. – URL: <https://znanium.com/catalog/product/2069273> (дата обращения: 31.05.2023)

Индивидуальное задание по учебной практике

Студент Бабкина Олеся Олеговна
ФИО

обучающийся на 2 курсе по специальности 09.02.07 «Информационные системы и программирование» прошел учебную практику в объеме 108 часов с «15» мая 2023 г. по «3» июня 2023 г. в организации ФГБОУ ВО "ВВГУ", Академический колледж, г. Владивосток
наименование организации, юридический адрес

Виды и объем работ в период производственной практики

№ п/п	Вид работ	Кол-во часов
1	Установочная лекция, инструктаж по охране труда и технике безопасности, распределение по рабочим местам	4
2	Ознакомиться с документацией и перечнем заданий на учебную практику	2
3	Разработать план мероприятий для этапов разработки, тестирования программного модуля.	4
4	Разработка программного модуля: - провести анализ требований к программным модулям по предложенной документации учебной практики; - разработать алгоритм и структуру данных; - программировать (кодировать) модуль; - шлифовать текст модуля; - проверить модуль; - транслировать модуль.	42
5	Разрабатывать тестовые наборы (пакеты) и сценарии программного средства для программного модуля.	20
6	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования: - инспектировать программный код на предмет соответствия стандартам кодирования, разработанного программного модуля; - инспектировать функциональное тестирование тестовых наборов (пакетов) и сценариев программных средств; - инспектировать тестирование интеграции тестовых наборов (пакетов) и сценариев программных средств.	24
7	Обобщить материалы практики, оформить отчет и необходимые документы по практике.	12

Дата выдачи задания «13» мая 2023 г.

Срок сдачи отчета по практике «3» июня 2023 г.

Подпись руководителя практики



/Атабаева Д.А., преподаватель АК ВВГУ

АТТЕСТАЦИОННЫЙ ЛИСТ

Студент Бабкина Олеся Олеговна

ФИО

обучающийся на 2 курсе по специальности 09.02.07 «Информационные системы и программирование» прошел учебную практику в объеме 108 часов с «15» мая 2023 г. по «3» июня 2023 г.

в организации ФГБОУ ВО «ВВГУ», Академический колледж, г. Владивосток

наименование организации, юридический адрес

В период практики в рамках осваиваемого вида профессиональной деятельности выполнял следующие виды работ:

Вид профессиональной деятельности	Код и формулировка формируемых профессиональных компетенций	Виды работ, выполненных обучающимся во время практики в рамках овладения компетенциями	Качество выполнения работ (отлично, хорошо, удовлетворительно, неудовлетворительно)
ПМ.01 Осуществление интеграции программных модулей	ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.	Ознакомиться с документацией и перечнем заданий на практику. Разработать план мероприятий для этапов разработки, тестирования программного модуля. Провести анализ требований к программным модулям по предложенной документации практики	<i>отлично</i>
	ПК 2.2 Выполнять интеграцию модулей в программное обеспечение.	Разработать план мероприятий для этапов проектирования и разработки информационной системы. Разработать алгоритм и структуру данных. Интегрировать готовый модуль в программное обеспечение.	<i>отлично</i>
	ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств.	Программировать (кодировать) модуль. Шлифовать текст модуля. Проверить и транслировать программный модуль.	<i>отлично</i>
	ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	Разрабатывать тестовые наборы (пакеты) и сценарии программного средства для программного модуля.	<i>отлично</i>
	ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	Инспектировать программный код на предмет соответствия стандартам кодирования, разработанного программного модуля. Инспектировать функциональное тестирование тестовых наборов (пакетов) и сценариев программных средств. Инспектировать тестирование интеграции тестовых наборов (пакетов) и сценариев программных средств	<i>отлично</i>
<i>Итоговая оценка по ПМ.01 Осуществление интеграции программных модулей</i>			<i>отлично</i>

Заключение об уровне освоения обучающимися профессиональных компетенций:

освоены на продвинутом уровне

(освоены на продвинутом уровне / освоены на базовом уровне /
освоены на пороговом уровне / освоены на уровне ниже порогового)

Дата 3 июня 2023 г.

Оценка за практику *отлично*

Руководитель практики от предприятия


подпись

Атабаева Д. А.
Ф.И.О.

М.П.

ДНЕВНИК

прохождения учебной практики

Студент (ка) Бабкина Олеся Олеговна

Фамилия Имя Отчество

Специальность 09.02.07 «Информационные системы и программирование»

Группа СО-ИП-21-1

Место прохождения практики ФГБОУ ВО «ВВГУ», Академический колледж, г. Владивосток

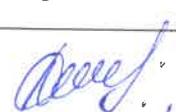
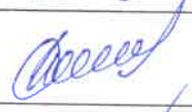
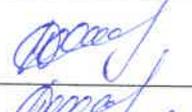
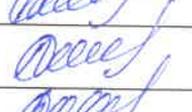
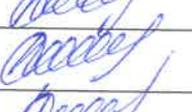
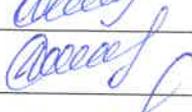
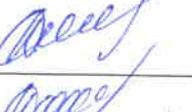
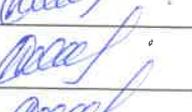
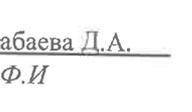
Сроки прохождения с 15.05.2023 г. по 03.06.2023 г.

Инструктаж на рабочем месте

«15» мая 2023 г
дата


подпись

Атабаева Д.А.
Ф.И.О. инструктирующего

Дата (период)	Описание выполнения производственных заданий (виды и объем работ, выполненных за день)	Оценка	Подпись руководителя практики
15.05.2023	Прохождение инструктажа по охране труда и технике безопасности, распределение по рабочим местам, установочная лекция.	5	
16.05.2023	Ознакомление с документацией и перечнем заданий на учебную практику. Разработка плана мероприятий для этапов разработки, тестирования программного модуля.	5	
17.05.2023	Проведение анализа требований к программным модулям по предложенной документации учебной практики.	5	
18.05.2023	Разработка алгоритма и структуры данных.	5	
19.05.2023 - 22.05.2023	Программирование модуля.	5	
23.05.2023	Шлифование текста модуля.	5	
24.05.2023 – 25.05.2023	Проверка модуля.	5	
26.05.2023	Транслирование модуля.	5	
27.05.2023	Разработка тестового набора и сценария программного средства для программного модуля.	5	
29.05.2023 – 30.05.2023	Инспектирование программного кода на предмет соответствия стандартам кодирования, разработанного программного модуля.	5	
31.05.2023	Инспектирование функционального тестирования текстовых наборов и сценариев программных средств.	5	
01.06.2023	Инспектирование тестирования интеграции текстовых наборов и сценариев программных средств.	5	
02.06.2023 г. – 03.06.2023 г.	Обобщение материалов практики, оформление отчета и необходимых документов по практике.	5	

Руководитель практики

М.П.


подпись

Атабаева Д.А.
Ф.И

Характеристика деятельности студента
Бабкиной Олеси Олеговны
 группы СО-ИП-21-1 при прохождении учебной практики

Код	Наименование компетенции	Основные показатели оценки результата	***Уровень (низкий, средний, высокий)
ПК 2.1	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.	Способность оформлять требования к программным модулям по предложенной документации	<i>высокий</i>
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение.	Способность организовывать контроль заданной интеграции модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов	<i>высокий</i>
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств.	Способность управлять процессом разработки программного модуля с использованием инструментальных средств. Модифицировать отдельные модули информационной системы. Способность организовывать постобработку данных	<i>высокий</i>
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	Способность определить требования к тестовым наборам (пакеты) для программного модуля. Способность определить требования к разработке тестовых сцен	<i>высокий</i>
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	Способность выполнить контроль разработанных программных модулей на предмет соответствия стандартам кодирования	<i>высокий</i>

Руководитель практики (от организации) _____

Атабаева Д.А.

Атабаева Д.А.

М.П.

ХАРАКТЕРИСТИКА

о прохождении учебной практики студента

Студент Бабкина Олеся Олеговна

ФИО студента

СО-ИП-21-1

группа

проходил практику с «15» мая 2023 г. по «03» июня 2023 г.

на базе ФГБОУ ВО «ВВГУ», Академический колледж, г. Владивосток

название предприятия

в подразделении _____

название подразделения

За период прохождения практики студент посетил 18 дней, из них по уважительной причине отсутствовал дней, пропуски без уважительной причины составили дней.

Студент соблюдал и правила техники безопасности.

Отмечены следующие нарушения трудовой дисциплины и/или правил техники безопасности:

Студент не справился со следующими видами работ:

За время прохождения практики студент показал, что умеет планировать и организовывать собственную деятельность, способен налаживать взаимоотношения с другими, имеет хороший уровень культуры поведения, умеет работать в команде, высокая степень сформированности умений в профессиональной деятельности.

В отношении выполнения трудовых заданий проявил такие качества, как ответственность и внимательность.

В рамках дальнейшего обучения и прохождения производственной практики студенту можно порекомендовать: углублять свои знания в области информационных технологий, а также самостоятельно осваивать новые инструменты и технологии разработки программных средств

Преподаватель АК ВВГУ

Должность наставника/куратора


подпись

Атабаева Д.А.

И.О. Фамилия

М.П.