

УДК 004.021

*Петрова Н.К., Кандидат физико-математических наук,  
Доцент кафедры «Информатика и информационно-управляющие  
системы»*

*Казанский Государственный Энергетический Университет  
Россия, г. Казань*

*Копунова Э., Ширмамедова З., Злыгостев Д.*

*Студенты*

*1 курс, факультет «Прикладная информатика»*

*Институт цифровых технологий и экономики*

*Казанский Государственный Энергетический Университет*

*Россия, г. Казань*

## **РАЗРАБОТКА WINDOWS-ПОДОБНОГО ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ РАБОТЫ С ТРЁХМЕРНЫМИ ВЕКТОРАМИ**

*Аннотация:* В математике часто встречаются задачи, где используются многомерные векторы. Обычно для ускорения операций над ними с помощью компьютера создаются массивы и процесс обработки сводится к большому количеству операторов Call, вызывающих разные процедуры. При этом список параметров в подобного рода операциях бывает достаточно длинными, что принципиально усложняет работу. В данном исследовании мы, во-первых, ввели пользовательский тип vector, что позволило принципиально упростить обращение к функциям и процедурам, а, во-вторых, разработали пользовательскую форму, которая предоставляет удобные средства для выполнения простейших операций над трёхмерными векторами.

*Ключевые слова:* windows-подобный пользовательский интерфейс, трёхмерный вектор, VBA, математические операции.

**Annotation:** *Mathematicians often encounter problems where multidimensional vectors are used. Usually, to speed up operations on them with the help of a computer, arrays are created and the processing process is reduced to a large number of Call operators that call different procedures. At the same time, the list of parameters in such operations is quite long, which essentially complicates the work. In this study, we first introduced a user-type vector, which made it possible to simplify fundamentally the access to functions and procedures, and secondly, we developed a custom form that provides convenient tools for performing simple operations on three-dimensional vectors.*

**Key words:** *windows-like user interface, three-dimensional vector, VBA, mathematical operations.*

## **Введение**

VBA – не просто стандартный макроязык приложений Microsoft Office: он используется для увеличения функциональных возможностей приложения, в котором применяется. К примеру, можно добавить собственное меню или функцию к встроенным возможностям Excel, или создать своего мастера, чтобы упростить создание презентаций PowerPoint.

Кроме расширения возможностей приложений, VBA позволяет действовать с данными, вывести или изменить их из других применяющих VBA прикладных программ, в частности, Visio или AutoCad: можно открыть базу данных Access, прочесть данные из таблицы и интегрировать ее с рабочим листом Excel, имеется возможность изменить внешний вид рисунка Visio, который создан на основе информации из документа Word. Совместно применяя VBA с иными приложениями, например, программами Microsoft Office, можно находить решение очень сложных задач.

Полный набор средств VBA открывается пользователю только тогда, когда он начинает изучать иерархию предметов приложений, в которых разрабатывается программа: при употреблении VBA и модели объектов имеется возможность управлять этими приложениями.

Также при помощи VBA можно делать математические расчеты. В частности, операции над векторами  $\vec{a}=x\vec{i}+y\vec{j}+z\vec{k}$ . В данной статье рассмотрена реализация основных кодов для операций над данными векторами.

## 1. Введение пользовательского типа и реализация основных кодов для операций над векторами $\vec{a}=x\vec{i}+y\vec{j}+z\vec{k}$ .

*Public Type Vector*

*x As Single*

*y As Single*

*z As Single*

*End Type*

Поля этого типа представляют собой прямоугольные координаты вектора.

Благодаря введению нового типа, прототипы программ существенно упрощаются. Например, функция сложения двух векторов записывается следующим образом:

*Public Function Add(a1 As Vector, a2 As Vector) As Vector*

При такой записи нет необходимости передавать массивы координат и их размеры. Аналогично выглядят и прототипы других функций. В целом, проект содержит следующие операции:

- Функция инициализации вектора.
- Процедура печати вектора в окне сообщений
- Функции сложения, вычитания векторов и умножения вектора на константу.
- Функция для вычисления скалярного произведения векторов.
- Функция для вычисления векторного произведения векторов.

## 2. Разработка пользовательской формы.

Форма, соответствующая описанию работы с трехмерными векторами, имеет следующий вид:

### Назначение основных кнопок формы:

#### 1) Для кнопки «Input»:

сначала проверяем содержимое текстовых полей, если в поле введена не цифра, то появляется соответствующее сообщение об ошибке, и пользователь может исправить введенный символ; далее происходит инициализация полей вектора; после занесения вектора в оперативную память, поля очищаются от введенных значений.

#### 2) Для кнопки «Output»:

Введенные векторы выводятся в соответствующую надпись Label, которая становится видимой на форме.

#### 3) Для кнопки «Exit»:

Код предназначен для выхода из приложения.

#### 4) Для кнопки «Calculate»:

Код позволяет выполнить одну из 5 операций: сложение векторов, вычитание, умножение на константу, скалярное и векторное произведения- выбор которой реализуется через переключатель.

Для каждой выбранной операции сначала проверяются введенные значения

индексов(должны быть числовые), по завершению операции пользователь может решить: выводить или не выводить получившийся вектор в поле Label.

### 3. Разработка контрольного примера.

Для отладки программы необходимо разработать контрольный пример, сравнение с результатами которого позволит определить корректность работы программы и её полноту. Мы рассмотрели следующие варианты векторов и операций над ними:

$$\bar{a}_1=5i+21j-7k$$

$$\bar{a}_2=7i-7j+10k$$

$$\bar{a}_3=\bar{a}_1+\bar{a}_2=12i+14j+3k$$

$$\bar{a}_4=\bar{a}_3-\bar{a}_2=5i+21j-7k$$

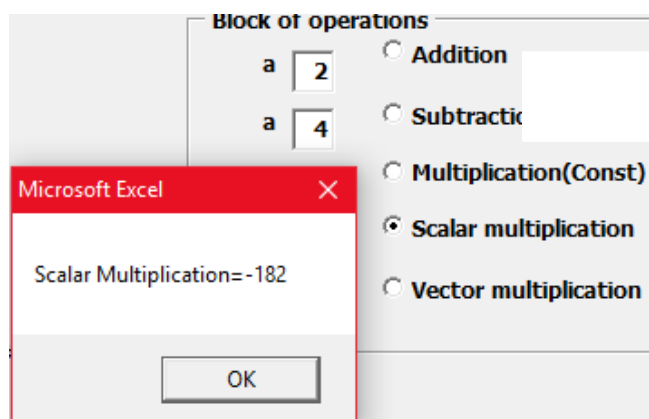
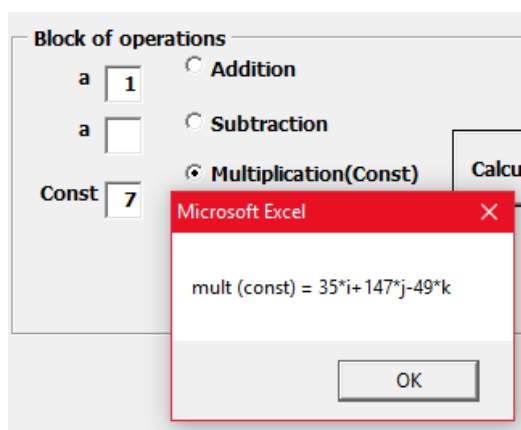
$$\bar{a}_1 \times (7)=35i+147j-49k$$

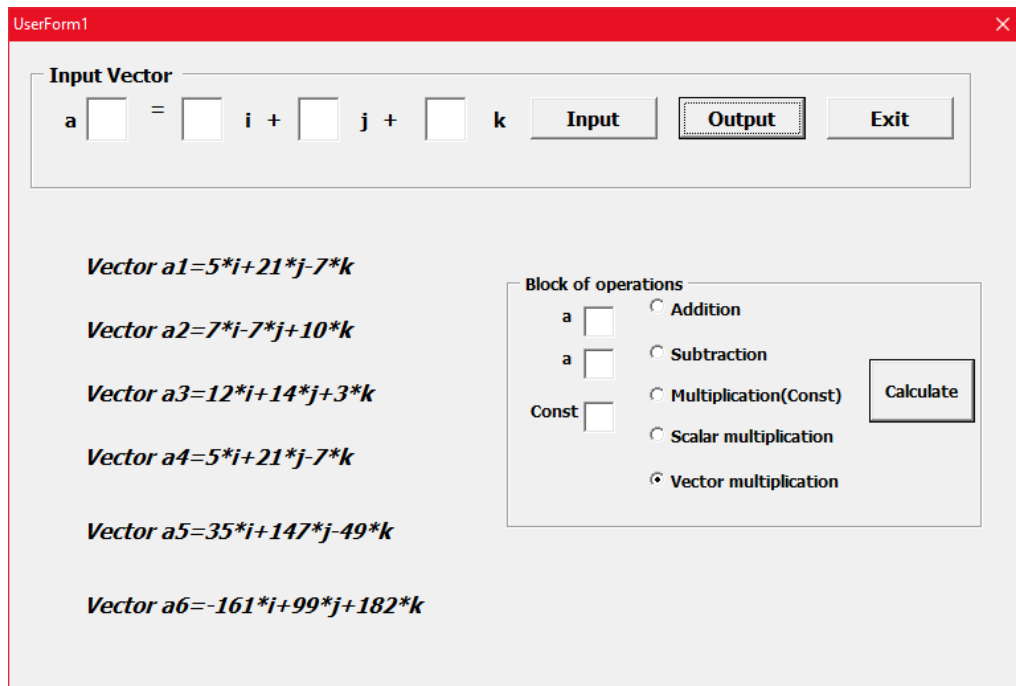
$$\bar{a}_2 \times \bar{a}_4=7 \times 5-7 \times 21+10 \times (-7)=-182$$

$$[\bar{a}_3 \times \bar{a}_4]=\begin{vmatrix} 14 * (-7) - 21 * 3 = -161 \\ 12 * (-7) - 3 * 5 = -99 \\ 12 * 21 - 14 * 5 = 182 \end{vmatrix} = -161i-99j+182k$$

### 4. Результаты работы программы

В поля формы были введены данные для векторов из контрольного примера. Результаты операций выводились как в поля *Label* формы, так, параллельно, и в окна сообщений. Следующая серия скриншотов демонстрирует последовательность работы с программой через созданную форму.





## 5. Заключение

Результаты, полученные программой, полностью воспроизводят контрольный пример, что свидетельствует о корректной работе программы. Работа с полями формы позволяет наглядно представлять не только результаты, но и исходные данные, выбирать произвольную последовательность выполнения операций. Проект готов к применению

## 6. Список литературы

1. Гарнаев А. Ю. Самоучитель VBA. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2004. – 560 с.
2. Зарипова Р.С. Среды разработки программных приложений как составная часть информационных технологий в подготовке технических специалистов /Р.С. Зарипова, Р.А. Ишмуратов / NovaUm.Ru. – 2018. – №11. – С.28-29.
3. Петрова Н.К. Разработка элементов пользовательского интерфейса при работе с электронными документами средствами языка Visual Basic. / Петрова Н.К., Рябова О.В., Зарипова Р.С. / Методические указания к практ. работам. Казань, КГЭУ, 2017 – 55 стр.