

# ИССЛЕДОВАНИЕ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ НА ЯЗЫКЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ VISUAL BASIC FOR APPLICATIONS

Информатика

**Буленков К. Д.**

*г. Тверь, Тверской лицей, 9 класс*

*Научный руководитель: Наумова А.И., г. Тверь, преподаватель информатики  
высшей категории, Тверской лицей*

## **Введение**

Данная статья является реферативным изложением основной работы. Полный текст научной работы, иллюстрации и иные дополнительные материалы доступны на сайте XXI Международного конкурса научно-исследовательских и творческих работ учащихся “Старт в науке” по ссылке: <https://school-science.ru/21/4/58200>.

В данной работе представлен материал по разработке проекта “Построение логической схемы триггера” на языке объектно-ориентированного программирования Visual Basic for Applications в табличном процессоре MS Excel 2007 и MS Excel 2016.

*Цель* данной работы заключается в том, чтобы получить *дополнительные знания* по работе с макросами и провести анализ полученных результатов.

*Задача* состоит в том, чтобы подобрать *соответствующий материал* с последующей систематизацией, обобщением и иллюстрацией текста.

Работа состоит из двух частей: *описательной* (даны определение объектно-ориентированного языка, характеристика основных этапов разработки информационных моделей) и *практической* (приведён пример разработки проекта построения логической схемы триггера с последующей обработкой данных на компьютере *двумя способами*).

## **Объектно-ориентированное программирование**

Объектно-ориентированное программирование (ООП) – методология программирования, основанная на представлении программы в виде

совокупности взаимодействующих объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Идеологически, ООП – подход к программированию как к *моделированию информационных объектов*, решающий на новом уровне основную задачу структурного программирования: структурирование информации с точки зрения управляемости, что существенно улучшает управляемость самим процессом моделирования [5].

### **Система объектно-ориентированного программирования**

Системы объектно-ориентированного программирования дают возможность *визуализировать процесс* создания графического интерфейса разрабатываемого приложения, то есть позволяет создавать объекты и задавать значения их свойств с помощью диалоговых окон системы программирования.

Взаимодействие программных объектов между собой и их изменения описываются с помощью программного кода. Создание программного кода в объектно-ориентированном программировании базируется на использовании алгоритмических структур различных типов (линейной, ветвления, цикла), исполнителями которых выступают программные объекты [3].

### **Язык объектно-ориентированного программирования**

#### **Visual Basic for Applications (VBA)**

Язык объектно-ориентированного программирования *Visual Basic for Applications (VBA)* предназначен для разработки приложений в среде *Microsoft Office*. Интерфейс интегрированной среды разработки VBA аналогичен интерфейсу интегрированной среды разработки *Visual Basic*. Запуск среды разработки VBA осуществляется из любого приложения, входящего в *Microsoft Office*. Окно среды разработки VBA представляет собой стандартное окно приложения, в котором могут быть открыты с помощью пунктов меню окна среды: *Панель инструментов*, окно *Конструктор форм*, *Свойства объекта*, *Программный код*, *Просмотр объектов* и *Проводник проекта*.

Последовательность действий пользователя будет записана в форме *макроса* (стандартного программного модуля на языке VBA).

С помощью VBA можно создавать проекты с *графическим интерфейсом*.

В данной работе рассмотрена разработка проекта на языке VBA в приложении Microsoft Office Excel; интерактивный диалог пользователя будем осуществлять с помощью графического интерфейса, а вычисления – с использованием возможностей электронных таблиц [3].

### **Основные этапы разработки и исследования моделей на компьютере**

Использование компьютера для исследования информационных моделей различных объектов и систем позволяет *изучить их изменения* в зависимости от значения тех или иных параметров. Процесс разработки моделей и их исследование на компьютере можно разделить на *несколько основных этапов*.

**Описательная информационная модель.** На *первом этапе* исследования объекта или процесса обычно строится *описательная информационная модель* на естественном языке. Такая модель выделяет *существенные*, с точки зрения целей проводимого исследования, свойства (параметры) объекта, а несущественными свойствами пренебрегает.

**Формальная модель.** На *втором этапе* создаётся *формализованная модель*, т.е. описательная информационная модель записывается с помощью какого-либо *формального* языка. В такой модели с помощью формул, уравнений, неравенств и т. д. фиксируются формальные соотношения между исходными и искомыми величинами, а накладываются ограничения на допустимые значения этих величин.

Однако далеко не всегда удаётся найти формулы, явно выражающие искомые величины через исходные. В таких случаях используются приближённые математические методы, позволяющие получать *результаты с заданной точностью*.

**Компьютерная модель.** На *третьем этапе* необходимо *формализованную информационную модель преобразовать в компьютерную модель*, т. е. выразить её на *понятном для компьютера языке*. Существуют два пути решения этой задачи:

- создание проекта на одном из языков программирования;

- построение компьютерной модели с использованием некоторого приложения, например электронных таблиц.

В процессе создания компьютерной модели полезно разработать *удобный графический интерфейс*, который позволяет *визуализировать* формальную модель, а также *реализовать интерактивный диалог* человека с компьютером на этапе исследования модели.

**Компьютерный эксперимент.** *Четвёртый этап* исследования информационной модели состоит в проведении *компьютерного эксперимента*. Если компьютерная модель существует в виде программы на одном из языков программирования, её нужно запустить на выполнение и получить результаты.

Если компьютерная модель исследуется в приложении, например в электронных таблицах, можно провести сортировку или поиск данных, построить диаграмму или график и т. д.

**Анализ полученных результатов и корректировка исследуемой модели.** *Пятый этап* представляет собой анализ полученных результатов и корректировку исследуемой модели. В случае отличия результатов, полученных при исследовании информационной модели, от измеренных параметров реальных объектов можно сделать вывод, что на предыдущих этапах построения модели были допущены ошибки или неточности.

Например, при построении описательной качественной модели могут быть неправильно отобраны существенные свойства объектов, в процессе формализации могут быть допущены ошибки в формулах и т. д. В таких случаях необходимо провести корректировку модели, причём уточнение модели *может проводиться многократно, пока анализ результатов не покажет их соответствие изучаемому объекту*.

**Визуализация формальных моделей.** В процессе исследования формальных моделей часто производится их *визуализация*. Для визуализации алгоритмов используются *блок-схемы*, пространственных соотношений параметров объектов – *чертежи*, моделей электрических цепей – *электрические схемы*. При визуализации формальных моделей с помощью анимации может

отображаться динамика процесса, производится построение графиков изменения величин и т. д.

В настоящее время широкое распространение получили *компьютерные интерактивные визуальные модели*. В таких моделях исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели [1].

### **Разработка проекта “Построение логической схемы триггера”**

#### **Содержательная постановка задачи**

При изучении *базовых логических элементов* компьютера (сумматор, триггер) целесообразно использовать *компьютерные модели*. Такие модели позволяют *визуализировать процесс преобразования логических значений входных сигналов в значения выходных сигналов*.

#### **Формальная модель**

Важнейшей структурной единицей оперативной памяти компьютера, а также внутренних регистров процессора является триггер. Триггер может находиться *в одном из двух устойчивых состояний*, что позволяет запоминать, хранить и считывать *1 бит* информации.

Триггер можно построить из двух логических элементов ИЛИ и двух элементов НЕ. При этом выход первого элемента НЕ соединён со входом второго элемента ИЛИ, а выход второго элемента НЕ соединён со входом первого элемента ИЛИ. Триггер имеет установочный вход S (от англ. set - установка) и вход сброса R (от англ. reset - сброс), а также выход Q (*рис.1*).

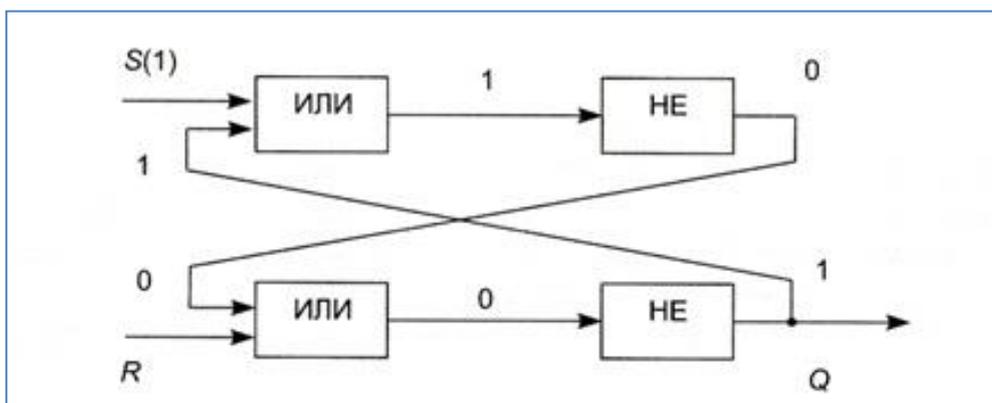


Рис. 1. Логическая схема триггера

Для записи 1 бита на вход S (установочный) подаётся сигнал 1. Последовательно рассмотрев по логической схеме прохождение сигнала, видим, что на выходе триггера Q устанавливается 1. Триггер переходит в это состояние и будет находиться в нём и после того, как сигнал на входе S исчезнет. Таким образом, триггер будет устойчиво хранить 1 бит.

Для того, чтобы сбросить бит данных и подготовиться к приёму нового бита, подаётся сигнал 1 на вход R (сброс), после чего триггер возвратится к исходному “нулевому” состоянию.

Подача на оба входа логической единицы может привести к неоднозначному результату, поэтому такая комбинация входных сигналов запрещена (табл. 1) [2].

Таблица 1. Таблица состояний входов и выходов триггера

Входы		Выход Q
S	R	
0	0	Q
1	0	1
0	1	0

## Компьютерная модель

### Разработка графического интерфейса проекта

1. Войти в приложение MS Excel.
2. Для создания графического интерфейса проекта (рис.2) запустить систему объектно-ориентированного программирования Visual Basic for Applications (VBA) в MS Excel 2007 командой [Office – Параметры Excel – Основные – Показывать вкладку “Разработчик” на ленте - Visual Basic]; в MS Excel 2016 командой [Файл – Параметры – Настроить ленту – Разработчик – Visual Basic] или использовать комбинацию клавиш Alt + F11 – войти в VBA, Alt + Q – закрыть VBA.
3. В окно системы программирования добавить форму командой [Insert – UserForm].
4. Поместить на форму:
  - 2 текстовых окна TextBox1 и TextBox2 для ввода начальных логических значений на входах триггера.

- Две метки Label для вывода промежуточных логических значений.
  - Две метки Label для вывода состояний выходов триггера.
  - Четыре метки Label для обозначения составляющих триггер логических элементов.
  - Четыре метки Label для обозначения входов и выходов триггера.
  - Кнопку CommandButton1 для запуска событийной процедуры установки значения триггера и сброса значения триггера [4].
5. Сделаем внешний вид проекта более привлекательным. Для этого изменим свойства объектов, определяющие их внешний вид (вручную установить размер объекта, цвет фона, цвет шрифта, начертание и размер шрифта, выравнивание шрифта).

AutoSize – False;

BackColor – Palette – цвет фона;

ForeColor – Palette – цвет шрифта;

Font – шрифт (Times New Roman), начертание (полужирный), размер шрифта (10), набор символов (кириллический);

Caption – ввести название формы, меток и командной кнопки;

TextAlign – fmTextAlignCenter (выравнивание).

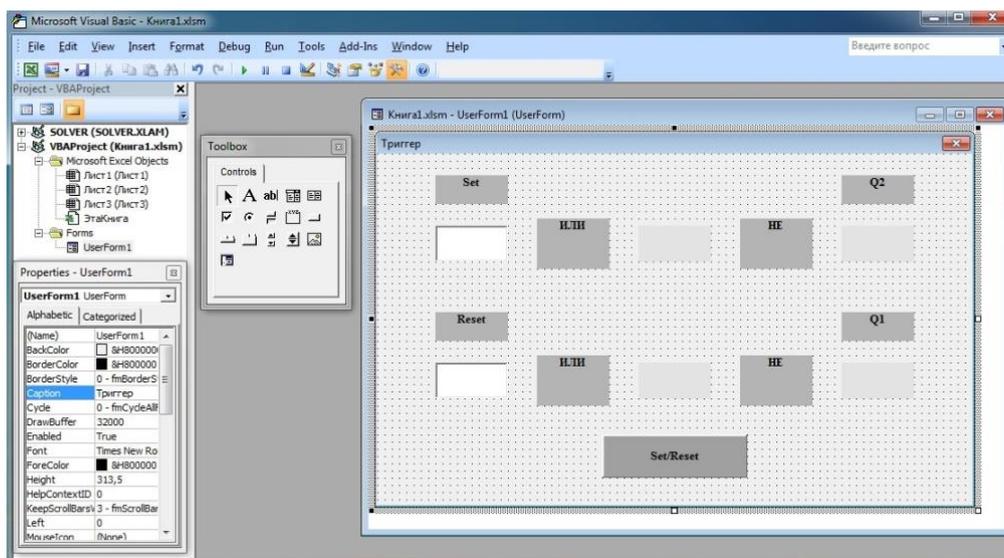


Рис. 2. Графический интерфейс проекта

**Событийные процедуры**

**Обработчик события: командная кнопка – CommandButton1**

Программный код можно создать *одним из двух способов*: с помощью использования *логических формул* приложения MS Excel или с помощью использования *логических операций* непосредственно в программном коде проекта.

### **Вариант 1: Логические формулы (Excel 2007).**

1. Заполнить столбец А (ввести названия переменных и логических функций).
2. Предусмотреть ввод логических значений аргументов в ячейки В1 и В2 (ввести двоичные значения).

Для ввода формул: *Выделить ячейку – Формулы – Логические – Выбрать функцию - Ввести значения <ОК>*.

3. В ячейку В3 ввести формулу =ИЛИ(В1;В2).  
В ячейку В4 ввести формулу =ЕСЛИ(В1=ИСТИНА;ИСТИНА;ЛОЖЬ)  
В ячейку В5 ввести формулу =НЕ(В4).  
В ячейку В6 ввести формулу =ИЛИ(В5;В2).  
В ячейку В7 ввести формулу =ЕСЛИ(В2=ИСТИНА;ИСТИНА;ЛОЖЬ).  
В ячейку В8 ввести формулу =НЕ(В7).

4. Программный код:

```
Private Sub CommandButton1_Click()
```

```
‘Ввести на входе начальные значения
```

```
Cells(1, 2) = TextBox1.Text
```

```
Cells(2, 2) = TextBox2.Text
```

```
‘Вывод промежуточных и конечных логических значений
```

```
Label5.Caption = Cells(4, 2)
```

```
Label9.Caption = Cells(5, 2)
```

```
Label6.Caption = Cells(7, 2)
```

```
Label10.Caption = Cells(8, 2)
```

```
End Sub
```

### **Вариант 2: Логические операции (Excel 2016).**

Программный код:

*‘Определить логические переменные*

Dim S, R, Or1, Or2, Not1, Not2 As Boolean

Private Sub CommandButton1\_Click()

*‘Ввести на входе начальные значения*

S = TextBox1

R = TextBox2

*‘Вычислить промежуточные и конечные логические значения*

Or1 = S Or Not2

If Or1 = 1 Then Not1 = 0 Else Not1 = 1

Or2 = Not1 Or R

If Or2 = 1 Then Not2 = 0 Else Not2 = 1

*‘Вывод промежуточных и конечных логических значений*

Label5 = Or1

Label9 = Not1

Label6 = Or2

Label10 = Not2

End Sub

### **Тестовые примеры (таблица состояний триггера)**

Тестовые примеры возьмём из таблицы состояний триггера (формальная модель). Исходные данные для выполнения компьютерного эксперимента в таблице выделены.

Таблица 1. Таблица состояний входов и выходов триггера

Входы		Выход Q
S	R	
0	0	Q
1	0	1
0	1	0

### **Компьютерный эксперимент**

Согласно таблице состояний входов и выходов триггера выполнение проекта проведём по *двум вариантам* подготовленных программных кодов. Исходное состояние триггера – “нулевое”.

### Выполнение проекта по варианту 1.

1. Запустить проект на выполнение. **Установить триггер:** ввести в поле TextBox1 (Set) значение **True**, а в поле TextBox2 (Reset) – значение **False**. Щёлкнуть по кнопке *Set/Reset*. Проследить за установкой значений на элементах триггера (рис. 3). Триггер хранит **True**.

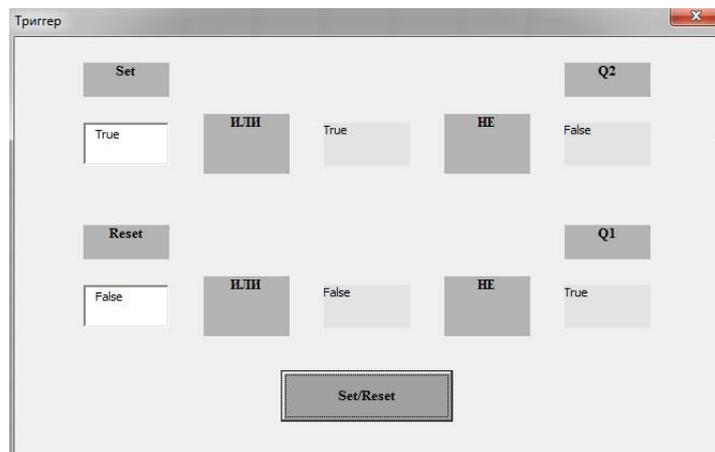


Рис.3. Триггер хранит **True** (на вход *Set* подан сигнал **True**)

2. **Сбросить триггер:** ввести в поле TextBox2 (Reset) значение **True**, а в поле TextBox1 (Set) – значение **False**. Щёлкнуть по кнопке *Set/Reset*. Проследить за установкой значений на элементах триггера (рис. 4). Триггер хранит **False**.

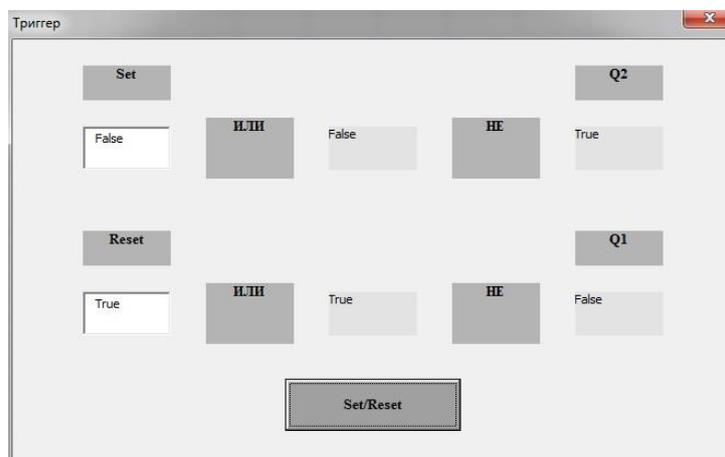


Рис. 4. Сброс триггера (на вход *Reset* подан сигнал **True**)

### Выполнение проекта по варианту 2.

1. Запустить проект на выполнение. **Установить триггер:** ввести в поле TextBox1 (Set) значение **1**, а в поле TextBox2 (Reset) – значение **0**. Щёлкнуть по кнопке *Set/Reset*. Проследить за установкой значений на элементах триггера (рис. 5). Триггер хранит **True**.

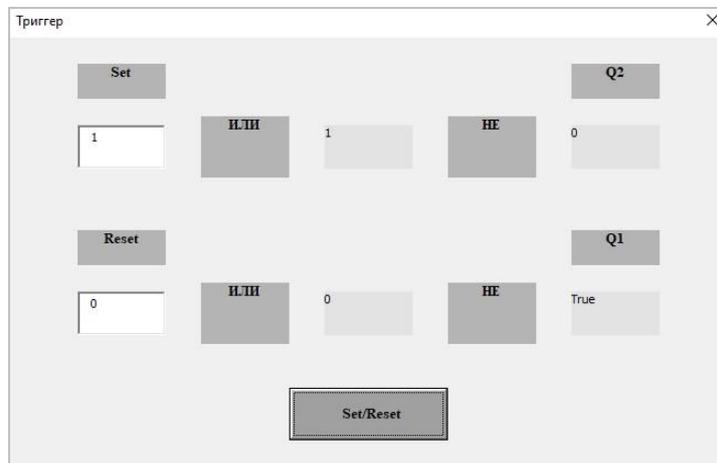


Рис. 5. Триггер хранит **True** (на вход *Set* подан сигнал **1**)

2. **Сбросить триггер:** ввести в поле TextBox2 (Reset) значение **1**, а в поле TextBox1 (Set) – значение **0**. Щёлкнуть по кнопке *Set/Reset*. Проследить за установкой значений на элементах триггера (рис. 6). Триггер хранит **False**.

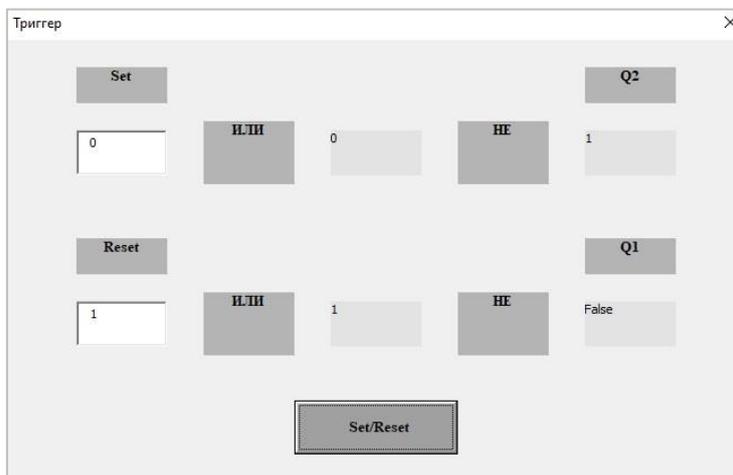


Рис. 6. Сброс триггера (на вход *Reset* подан сигнал **1**)

**Примечание:** Если при выполнении высвечивается сообщение об ошибке, нажать на кнопку *<Debug>*. На экране появится окно с ошибкой. Внести исправления и запустить проект вновь.

Для сохранения программного кода выполнить команду [кнопка Office (Файл) – Сохранить как... - Выбрать папку для сохранения документа – Имя файла (Книга1) – Тип файла (Книга Excel с поддержкой макросов) - Сохранить].

Для повторного запуска программного кода выполнить команды:

1. Войти в приложение MS Excel 2007 (2016);
2. Выполнить команду [кнопка Office (Файл) – Параметры – Центр управления безопасностью – Параметры центра управления безопасностью – Параметры макросов – Включить все макросы (не рекомендуется)] <ОК>;
3. Выполнить команду [кнопка Office (Файл) – Открыть – Книга1 – Открыть];
4. Войти в среду программирования Visual Basic for Applications: [Разработчик – Visual Basic];
5. По окончании работы выполнить команду [Отключить все макросы без уведомления].

### **Анализ полученных результатов**

При компьютерном выполнении проекта рассмотрены два варианта:

1 – программный код написан на основании значений в ячейках приложения MS Excel 2007, полученных в результате использования соответствующих схеме триггера логических формул;

2 – необходимые логические операции внесены непосредственно в программный код и исследование проведено в приложении MS Excel 2016.

На основании таблицы состояний триггера результаты, полученные при выполнении проекта по варианту 2, полностью соответствуют результатам, полученным при выполнении проекта по варианту 1.

При вводе в поле TextBox1 (Set) значение **1 (True)**, а в поле TextBox2 (Reset) – значение **0 (False)** триггер хранит **True**. При вводе в поле TextBox1 (Set) значение **0 (False)**, а в поле TextBox2 (Reset) – **1 (True)** триггер возвратится к исходному “нулевому” состоянию (рис.7).

<b>Выполнение проекта по варианту 1</b>	<b>Выполнение проекта по варианту 2</b>
---	---



Рис. 7. Результаты компьютерного эксперимента

### Заключение

В данной работе показана *разработка компьютерной модели* логического устройства триггера, которая позволяет *визуализировать процесс преобразования логических значений*.

Приведены примеры выполнения *двух программных кодов* в среде программирования Visual Basic for Applications для табличного процессора MS Excel 2007 и MS Excel 2016 с помощью логических формул приложения и с помощью использования логических операций непосредственно в программном коде. Проведён *сравнительный анализ планируемых и полученных результатов* компьютерного эксперимента.

В ходе разработки проекта получены дополнительные знания по теме “Логические основы компьютера”. Проведённый компьютерный эксперимент наглядно показывает *практическую значимость* проведённых исследований.

Visual Basic for Applications – объектно-ориентированный язык программирования, изучение которого начинается *в предпрофильных и профильных классах* общеобразовательных учреждений (факультативные занятия: Решение задач повышенной сложности) на примере *использования его основных объектов и их свойств*, что достаточно *наглядно* показано в данной работе.

#### **Список использованных источников и литературы**

1. Информатика и ИКТ. Профильный уровень : учебник для 11 класса / Н. Д. Угринович. – 2-е изд., испр. и доп. - М.: БИНОМ. Лаборатория знаний, 2009. – 308 с.: ил.
2. Информатика и ИКТ. Учебник для 10 класса / Н. Д. Угринович. - М.: БИНОМ. Лаборатория знаний, 2007. – 371 с.: ил. – (Профильный уровень)
3. Информатика и информационные технологии. Учебник для 10-11 классов / Н. Д. Угринович. – 3-е изд. - М.: БИНОМ. Лаборатория знаний, 2006. – 511 с.: ил.
4. Исследование информационных моделей. Элективный курс: Учебное пособие / Н. Д. Угринович – М.: БИНОМ. Лаборатория знаний, 2004. – 183 с.: ил.
5. Объектно-ориентированное программирование [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Объектно-ориентированное\\_программирование](https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование)

#### **Программное обеспечение**

1. Операционная система Windows 10
2. Приложение Microsoft Office Excel 2007
3. Приложение Microsoft Office Excel 2016
4. Приложение Visual Basic for Applications
5. Приложение Microsoft Office Word 2010