

# ИССЛЕДОВАНИЕ ВЕРОЯТНОСТНЫХ МОДЕЛЕЙ НА ЯЗЫКЕ VISUAL BASIC FOR APPLICATIONS

Информатика

**Погорелый С.К.**

*г. Тверь, Академическая гимназия Тверского государственного университета,  
11 класс*

*Научный руководитель: Наумова А.И., г. Тверь, преподаватель информатики  
высшей категории, Академическая гимназия Тверского государственного  
университета*

## **Введение**

Данная статья является реферативным изложением основной работы. Полный текст научной работы, иллюстрации и иные дополнительные материалы доступны на сайте XXVII Международного конкурса научно-исследовательских и творческих работ учащихся “Старт в науке” по ссылке: <https://school-science.ru/27/4/63692>.

В данной работе представлен материал по разработке проекта “Исследование вероятностных моделей на языке Visual Basic for Applications” в табличном процессоре MS Excel 2016. Тема достаточно *актуальная* и представляет *повышенный интерес для учащихся профильных классов*.

*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 вероятностных моделей, объектно-ориентированного языка программирования Visual Basic for Applications (VBA), который позволяет создавать *макросы* (стандартные программные модули), история его развития) и *практической* (приведён пример разработки и исследования проекта *моделирования игрального кубика на компьютере при максимальном количестве бросаний, равным 200 с использованием математической функции rnd()*).

## Вероятностные модели

### Определение вероятностных моделей

Вероятностные модели базируются на использовании *больших серий испытаний со случайными параметрами*, причём *точность* полученных результатов зависит от *количества проведённых опытов* [2].

Данный проект описывает *моделирование* средствами электронной таблицы Microsoft Excel и языка Visual Basic for Applications (VBA) *многократного бросания* игрального кубика с точки зрения *уменьшения неопределенности знаний* и *подсчета количества выпадений* того или иного числа на грани кубика. Результаты подсчета (меняющиеся в ходе проведения эксперимента) будем выводить в таблицу и диаграмму.

### Объектно-ориентированное программирование

#### 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 [2].

## **История развития языка VBA**

Язык программирования VBA (Visual Basic for Applications) прошел долгий путь эволюции, преобразуясь и адаптируясь под изменяющиеся потребности разработчиков и пользователей. Понимание развития этого языка поможет лучше оценить его потенциал и историческую значимость.

Основы VBA были заложены в 1960-х годах, когда был создан язык программирования BASIC. Разработанный для обучения программированию, BASIC отличался *простотой и доступностью*. Его главной целью было сделать программирование понятным для широкой аудитории.

С развитием *графических пользовательских интерфейсов* (GUI) в 1980-х и 1990-х годах, возникла необходимость в более мощных и гибких инструментах разработки. Microsoft выпустила Visual Basic (VB) в 1991 году, который объединил возможности BASIC с элементами дизайна пользовательского интерфейса.

- 1991 год: Выпуск первой версии Visual Basic;
- 1993 год: Введение Visual Basic for Applications, интегрированного в Microsoft Office;
- Обеспечение поддержки объектов, событий и элементов управления GUI.

С начала 1990-х годов VBA стал неотъемлемой частью пакета приложений Microsoft Office, что позволило ему получить широкое признание среди корпоративных пользователей и разработчиков. VBA позволял автоматизировать задачи, создавать макросы и разрабатывать сложные программные решения прямо внутри приложений Office.

*Сегодня VBA остается важным инструментом* для автоматизации работы в Microsoft Office и других приложениях. Несмотря на появление новых

технологий и языков программирования, VBA продолжает использоваться благодаря своей интеграции и простоте использования.

- Поддержка новых версий Microsoft Office;
- Адаптация к изменяющимся требованиям пользователей;
- Сохранение актуальности благодаря широкому сообществу разработчиков [4].

## **Исследование проекта “Моделирование бросания игрального кубика”**

### **Содержательная постановка задачи**

Согласно теории, при бросании шестигранного игрального кубика может наблюдаться *6 равновероятных событий*. Следовательно, с каждым бросанием *уменьшается неопределенность наших знаний в шесть раз*. Чем больше количество возможных событий, тем больше начальная неопределенность и соответственно тем больше количество информации будет содержать сообщение о результатах опыта [1].

### **Формальная модель**

1. Объединим ячейки B2 и C2 [Формат ячеек... – Выравнивание].
2. Введем текст в соответствующие ячейки и числа (1 – 6) в диапазон ячеек B4:B9.
3. В ячейке D2 будет выводиться *общее число “бросаний”* кубика, в ячейках C4:C9 – *общее количество “бросаний” по выпавшему числу*, в ячейках D4:D9 – *доля в % от значений в ячейках C4:C9* (формулы для расчета, например в ячейке D4:  $=C4*6/100$ , во всех последующих аналогично), в ячейке F4 будет выводиться *последнее число*, которое выпало на кубике (рис. 1).
4. Установить в ячейках необходимые параметры:
  - выравнивание текста *по центру*;
  - формат данных в ячейках D4:D9 – *процентный*.

	A	B	C	D	E	F
1						
2		Общее количество испытаний		0		
3		Число	Количество выпадений	Доля в %		Выпало
4		1	0	0,00%		0
5		2	0	0,00%		
6		3	0	0,00%		
7		4	0	0,00%		
8		5	0	0,00%		
9		6	0	0,00%		
10						

Рис. 1. Исходные данные в табличном процессоре

## Компьютерная модель

### Разработка графического интерфейса проекта

1. Войти в приложение MS Excel 2016.
2. Для создания графического интерфейса проекта (рис.2) запустить систему объектно-ориентированного программирования *Visual Basic for Applications (VBA)* 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 4 командные кнопки:
  - для запуска событийной процедуры (Бросить 1 раз);
  - для запуска событийной процедуры (Бросить 100 раз подряд);
  - для запуска событийной процедуры (Бросить 200 раз подряд);
  - для запуска событийной процедуры (Очистить).

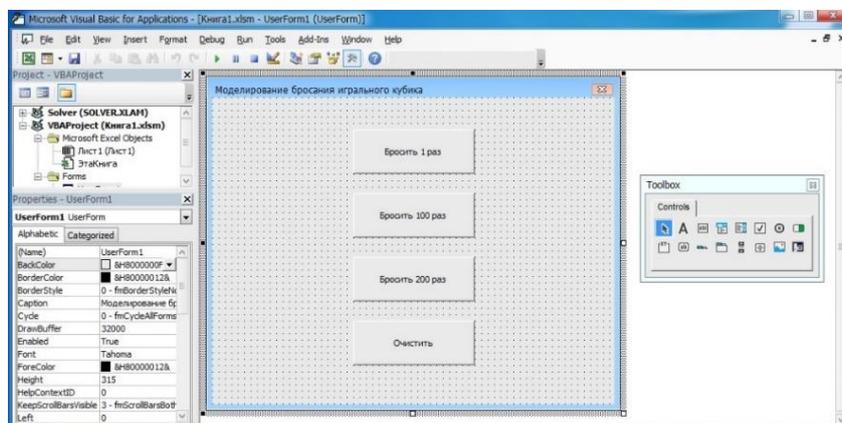


Рис. 2. Графический интерфейс проекта

5. Внешний вид проекта можно сделать 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) (рис. 3);

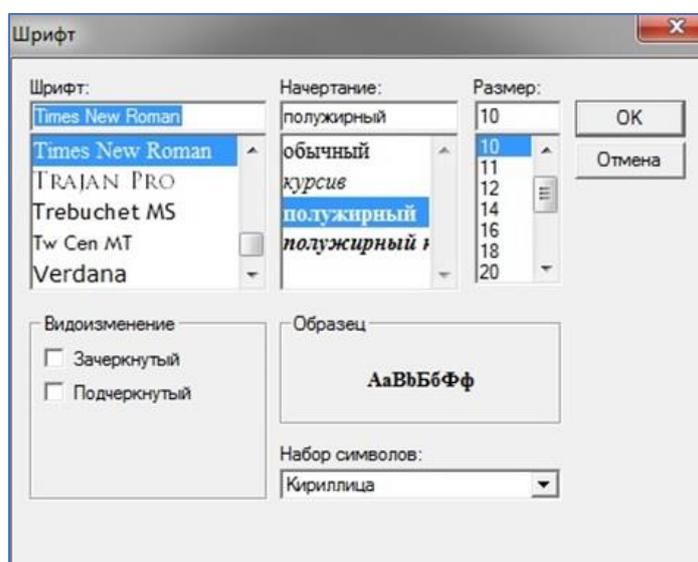


Рис. 3. Выбор шрифта, его начертания и размера

Caption – ввести название формы и командных кнопок.

### Создать событийные процедуры

Создать программный код для 1-й командной кнопки (CommandButton1),

щёлкнув по ней дважды:

```
Private Sub CommandButton1_Click()
```

```
'Выполнить одно бросание
```

```
Dim Выпало As Integer
```

```
Randomize
```

```
Выпало = Int(Rnd(1) * 6 + 1)
```

‘Определение стороны кубика

```
Range("F4").Select
```

ActiveCell.Value = Выпало

Select Case Выпало                    'Выбор ячейки в зависимости от стороны кубика

Case 1

Range("C4").Select

Case 2

Range("C5").Select

Case 3

Range("C6").Select

Case 4

Range("C7").Select

Case 5

Range("C8").Select

Case 6

Range("C9").Select

End Select

'Увеличить общее количество бросаний на 1

ActiveCell.Value = ActiveCell.Value + 1

Range("D2").Select

ActiveCell.Value = ActiveCell.Value + 1

End Sub

**Создать программный код для 2-й командной кнопки (CommandButton2),**

щёлкнув по ней дважды:

Private Sub CommandButton2\_Click()

'Выполнить 100 бросаний

Dim i As Integer

For i = 1 To 100

CommandButton1\_Click        'Вызов созданной ранее процедуры (одно бросание)

Next i

End Sub

**Создать программный код для 3-й командной кнопки (CommandButton3),**

щёлкнув по ней дважды:

```
Private Sub CommandButton3_Click()
```

```
'Выполнить 200 бросаний
```

```
Dim i As Integer
```

```
For i = 1 To 200
```

```
CommandButton1_Click      'Вызов созданной ранее процедуры (одно бросание)
```

```
Next i
```

```
End Sub
```

**Создать программный код для 4-й командной кнопки (CommandButton4),**

щёлкнув по ней дважды:

```
Private Sub CommandButton4_Click()
```

```
'Внести в рабочие ячейки нулевые значения
```

```
Range("F4").Select
```

```
ActiveCell.Value = 0
```

```
Range("C4").Select
```

```
ActiveCell.Value = 0
```

```
Range("C5").Select
```

```
ActiveCell.Value = 0
```

```
Range("C6").Select
```

```
ActiveCell.Value = 0
```

```
Range("C7").Select
```

```
ActiveCell.Value = 0
```

```
Range("C8").Select
```

```
ActiveCell.Value = 0
```

```
Range("C9").Select
```

```
ActiveCell.Value = 0
```

```
Range("D2").Select
```

```
ActiveCell.Value = 0
```

```
End Sub
```

## Компьютерный эксперимент

Запустить проект на выполнение, выполнив команду [Run – Run Module (F5)] (рис. 4).

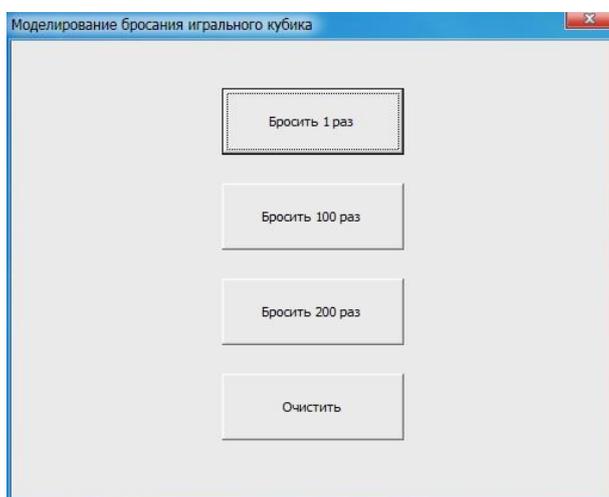


Рис. 4. Режим выполнения проекта

**Нажать на командную кнопку “Бросить 1 раз”**

	A	B	C	D	E	F
1						
2		Общее количество испытаний		1		
3		Число	Количество выпадений	Доля в %		Выпало
4		1	0	0,00%		6
5		2	0	0,00%		
6		3	0	0,00%		
7		4	0	0,00%		
8		5	0	0,00%		
9		6	1	6,00%		
10						

Рис. 5. Выполнение программного кода по командной кнопке CommandButton1

**Нажать на командную кнопку “Бросить 100 раз подряд”**

	A	B	C	D	E	F
1						
2		Общее количество испытаний		101		
3		Число	Количество выпадений	Доля в %		Выпало
4		1	17	102,00%		5
5		2	19	114,00%		
6		3	18	108,00%		
7		4	15	90,00%		
8		5	13	78,00%		
9		6	19	114,00%		
10						

Рис. 6. Выполнение программного кода по командной кнопке CommandButton2

### Нажать на командную кнопку “Бросить 200 раз подряд”

	A	B	C	D	E	F
1						
2		Общее количество испытаний		301		
3		Число	Количество выпадений	Доля в %		Выпало
4		1	39	234,00%		6
5		2	58	348,00%		
6		3	52	312,00%		
7		4	49	294,00%		
8		5	47	282,00%		
9		6	56	336,00%		
10						

Рис. 7. Выполнение программного кода по командной кнопке CommandButton3

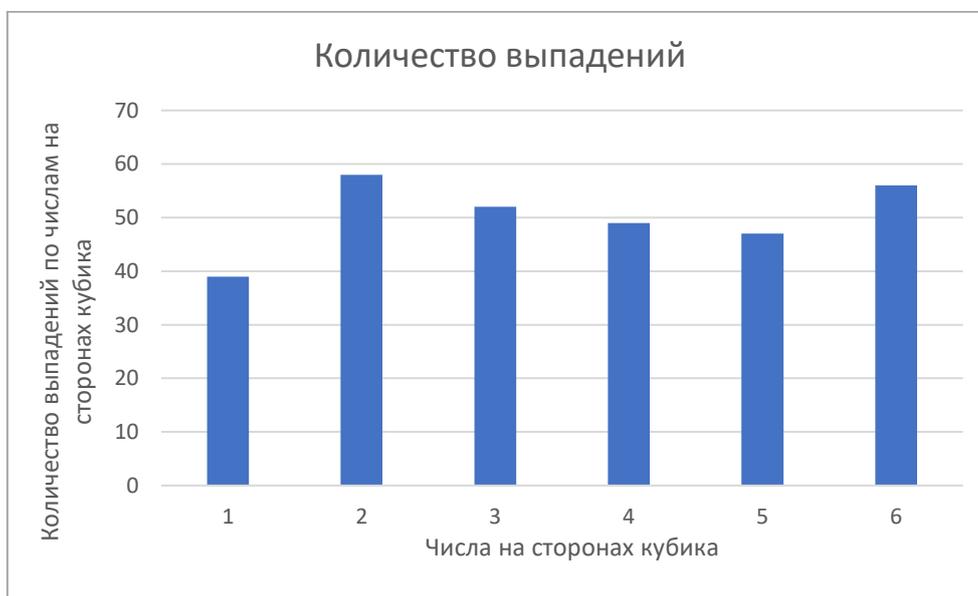


Рис. 8. Диаграмма полученных выпадений по рис. 7

### Нажать на командную кнопку “Очистить”

	A	B	C	D	E	F
1						
2		Общее количество испытаний		0		
3		Число	Количество выпадений	Доля в %		Выпало
4		1	0	0,00%		0
5		2	0	0,00%		
6		3	0	0,00%		
7		4	0	0,00%		
8		5	0	0,00%		
9		6	0	0,00%		
10						

Рис. 9. Выполнение программного кода по командной кнопке CommandButton4

**Примечание:** Если при выполнении программных кодов (рис.5-7) 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 *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 2016;
2. Выполнить команду [кнопка *Файл* – *Параметры* – *Центр управления безопасностью* – *Параметры центра управления безопасностью* – *Параметры макросов* – *Включить все макросы (не рекомендуется)*] <ОК>;
3. Выполнить команду [кнопка *Файл* – *Открыть* – *Книга1* – *Открыть*];
4. Войти в среду программирования Visual Basic for Applications: [*Разработчик* – *Visual Basic*] или использовать комбинацию клавиш Alt + F11;
5. По окончании работы выполнить команду [*Отключить все макросы без уведомления (п.2)*].

### **Анализ полученных результатов**

При компьютерном выполнении проекта рассмотрены *четыре варианта выполнения событийных процедур (программных кодов)*:

- 1-й программный код – для бросания 1 раз,
- 2-й программный код – для бросания 100 раз подряд,
- 3-й программный код – для бросания 200 раз подряд,
- 4-й программный код – очистить рабочие ячейки.

По первым трём программным кодам на основании представленной ниже таблицы (рис.10) можно сделать вывод:

*Чем больше количество проведённых бросаний, тем больше количество информации будет содержать сообщение о результатах проведённого эксперимента – сколько раз и какие грани выпали* (рис. 6, рис. 7).

Когда мы бросаем игральный кубик два (три, 101, 301) раза мы получаем информации в два (три, 101, 301) раза больше, чем при однократном бросании кубика.

Это свойство называют принципом **аддитивности** (сложения) [3].

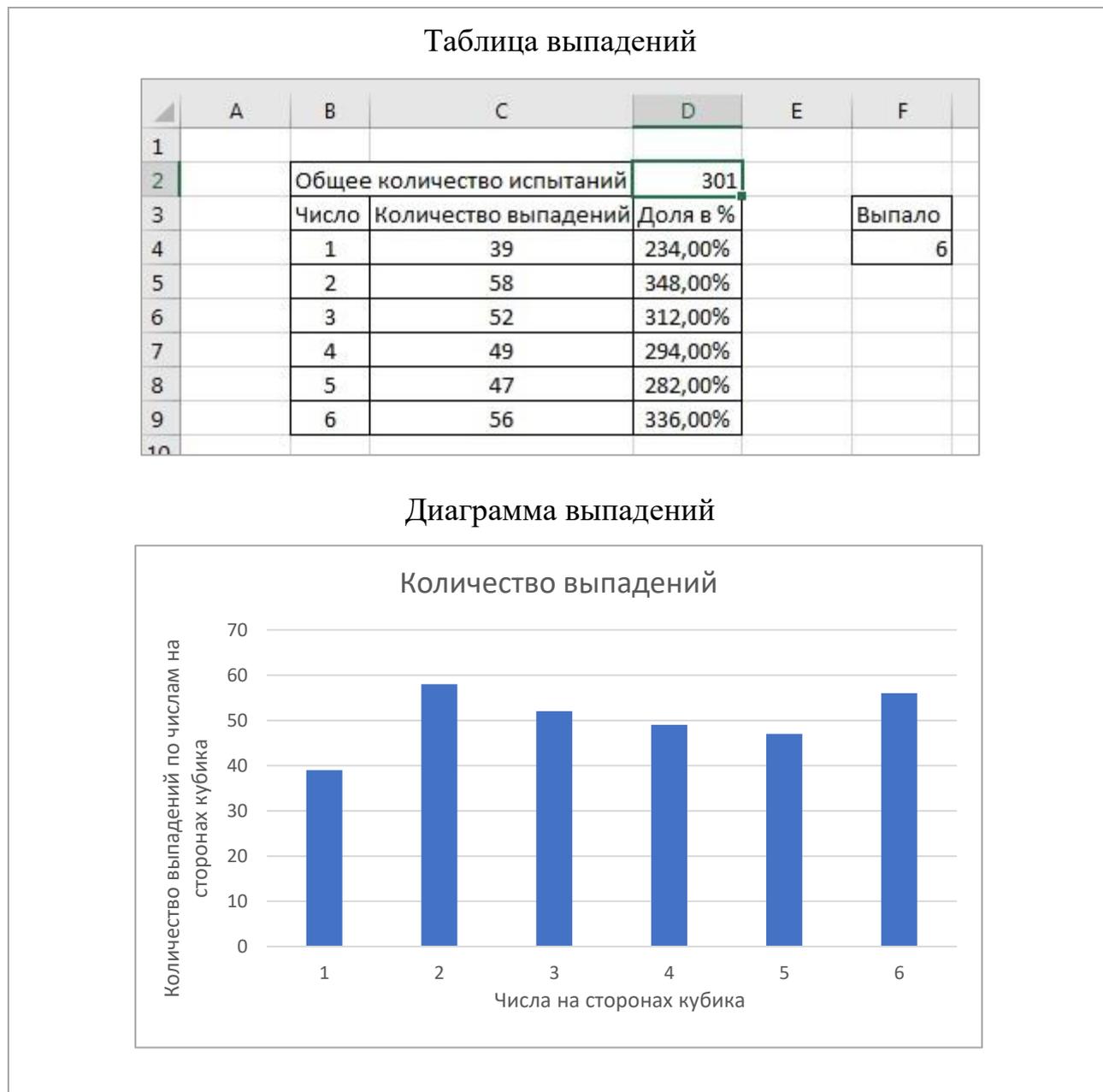


Рис. 10. Результаты компьютерного эксперимента по 3-й командной кнопке

### Заключение

В данной работе показана *разработка компьютерной модели бросания игрального кубика*, которая позволяет *визуализировать процесс получения числовых значений*.

Приведены примеры выполнения *четырёх программных кодов: Бросить 1 раз, Бросить 100 раз подряд, Бросить 200 раз подряд и Очистить* рабочие ячейки в среде программирования Visual Basic for Applications для табличного процессора MS Excel 2016. Проведён *сравнительный анализ полученных результатов* компьютерного эксперимента.

В ходе разработки проекта получены *дополнительные знания* по теме “Моделирование вероятностных моделей”. Проведённый компьютерный эксперимент наглядно показывает *практическую значимость* проведённых исследований.

Visual Basic for Applications – объектно-ориентированный язык программирования, изучение которого начинается *в 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.

#### **Программное обеспечение**

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

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

1. Журнал “ИНФОРМАТИКА” / Издательский дом “ПЕРВОЕ СЕНТЯБРЯ”, г. Москва, периодическое издание, № 6 / 2008 [Электронный ресурс]. Режим доступа: <https://inf.1sept.ru/index.php?year=2008&num=06>
2. Информатика и информационные технологии. Учебник для 10-11 классов / Н. Д. Угринович. – 3-е изд. - М.: БИНОМ. Лаборатория знаний, 2006. – 511 с.: ил.
3. Информатика. 11 класс (базовый и углубленный уровни) (в 2 частях): учебник. Ч. 1 / К.Ю. Поляков, Е.А. Еремин – М.: Бином. Лаборатория знаний, 2019. – 240 с.: ил.
4. Школа английского языка Skyeng: [Электронный ресурс]. Режим доступа: <https://skyeng.ru/magazine/wiki/it-industriya/chto-takoe-vba/>