**Департамент образования города Москвы**

**Государственное бюджетное образовательное учреждение**

**Школа № 1502 при МЭИ**

**Проектно-исследовательская работа**

**Модуль виртуальной сборки кубика Рубика**

**Выполнил:**

**Шпаковский Павел**

**ученик 10 И класса**

**Руководитель:**

**Буркатовская О.С.,**

**учитель информатики**

**Москва**

**2020**

Содержание

[Введение 3](#_Toc30943896)

[1 Методики сборки кубика Рубика и их базовые алгоритмы 4](#_Toc30943897)

[1.1 Написание формул и язык вращений 4](#_Toc30943898)

[1.2 Разделение сборки на характерные этапы 5](#_Toc30943899)

[2 Реализация модуля виртуальной сборки 5](#_Toc30943900)

[2.1 Средства разработки 5](#_Toc30943901)

[2.2 Составление алгоритма для сборки кубика Рубика 6](#_Toc30943902)

[2.3 Пользовательский интерфейс и 3D модель 6](#_Toc30943903)

[3 Анализ полученных результатов 7](#_Toc30943904)

[Заключение 8](#_Toc30943905)

[Список использованных источников 9](#_Toc30943906)

[Приложение 10](#_Toc30943907)

# Введение

В настоящее время стремительно набирает популярность такая головоломка, как кубик Рубика. Существуют и другие виды и формы данной головоломки, но классический кубик 3×3 является наиболее популярным. С недавних пор сборка головоломок на скорость стала видом спорта - спидкубингом. Для многих людей рекорды, поставленные на соревнованиях, являются чем-то недостижимым и непонятным, а некоторые пытаются изучить материал и научиться собирать данную головоломку.

Цель данного проекта - создать модуль виртуальной сборки кубика Рубика, который значительно упростит процесс обучения и сможет его визуализировать. Пользователь будет взаимодействовать с 3D моделью кубика, и обучаться основным алгоритмам сборки.

Для создания программы будет использоваться объектно-ориентированный язык программирования Java, преимуществом является его кроссплатформенность. Чтобы работать с графикой, будет использована встроенная библиотека JavaFX. Она предоставит доступ к 3D объектам и элементам пользовательского интерфейса. Непосредственное обучение пользователя будет основано на базовых алгоритмах, предназначенных для сборки этой головоломки.

При реализации проекта был поставлен ряд задач: исследовать теорию, связанную с методиками сборки кубика Рубика и выявить основные алгоритмы, наиболее простые для восприятия. Для создания программы необходимо ознакомиться с основами языка программирования, а далее разработать модуль для вычисления полного алгоритма сборки кубика Рубика из любого начального положения. Заключительным этапом работы будет создание пользовательского интерфейса для удобного взаимодействия с 3D моделью головоломки.

# Методики сборки кубика Рубика и их базовые алгоритмы

## 1.1 Написание формул и язык вращений

Язык вращений – это особые обозначения движений граней кубика, при помощи которых можно записать какой-либо алгоритм, решение или скрамбл (последовательность ходов, с помощью которой запутывают кубик).

F - front - фронтальная сторона

B - back - задняя сторона

L - left - левая сторона

R - right - правая сторона

U - up - верхняя сторона

D - down - нижняя сторона

y - весь куб вращается по часовой стрелке в горизонтальной плоскости (F превращается в L)

y' - весь куб вращается против часовой стрелки в горизонтальной плоскости (F превращается в R)

- Если написана только буква - выполняется поворот на 90 градусов по часовой стрелке.

- Если после буквы стоит штрих «'» - выполняется поворот на 90 градусов против часовой стрелки.

- Если после буквы стоит «2» - выполняется поворот на 180 градусов.

## 1.2 Разделение сборки на характерные этапы

* Сборка правильного креста.

Первый этап сборки кубика Рубика - сборка правильного креста на любой из сторон, для облегчения обучения будет использована белая сторона (см. Рис 1 Этапы сборки кубика Рубика).

* Сборка первого слоя.

Ориентирование угловых элементов на первом слое.

* Сборка второго слоя.

На данном этапе собирается второй слой. Для этого необходимо найти 4 ребра и поставить их на места между центрами второго слоя.

* Сборка желтого креста.

На этапе сборки желтого креста может быть всего 3 ситуации, решив которые, мы его получим.

* Сборка желтой стороны.

После того, как собран желтый крест, возможны 7 ситуаций, для каждой из которых придется запомнить алгоритм, чтобы собрать желтую сторону.

* Сборка углов верхнего слоя. Ориентирование угловых элементов третьего слоя.
* Сборка ребер верхнего слоя. Возможны 4 ситуации.

# 2 Реализация модуля виртуальной сборки

## 2.1 Средства разработки

Для создания программы используется язык программирования Java. Java - объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems. Изначально язык разрабатывался для бытовой электроники, но впоследствии был переименован и стал использоваться для написания апплетов, приложений и серверного программного обеспечения.

Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее. Средой разработки является IntelliJ IDEA, которая упрощает работу со стандартными библиотеками языка.

## 

## 2.2 Составление алгоритма для сборки кубика Рубика

Для реализации вращений и анализа головоломки были созданы классы “Cube”, который отвечает за взаимодействие с отдельными элементами кубика, и “CubeModel”, который содержит трехмерный массив из объектов класса “Cube” и основные функции для взаимодействия с этой моделью. Также для каждого этапа сборки существуют отдельные функции, каждая из которых возвращает строку, содержащую полный алгоритм для решения данного этапа. Итоговый алгоритм (формула из символов языка вращения) - это сумма строк всех этапов.

Проблема в том, что итоговый алгоритм содержит множество лишних движений, которые могут быть сведены к минимуму. Для этого используется одна из реализаций алгоритма RLE. Алгоритм RLE - алгоритм сжатия, заменяющий идущие подряд одинаковые символы парой (повторяющийся символ, количество повторений). Таким образом, на выходе получается оптимальный алгоритм сборки кубика Рубика из заданного начального положения.

## 2.3 Пользовательский интерфейс и 3D модель

Для повышения эффективности обучения пользователя необходим пользовательский интерфейс(GUI, см. Рис 2 Пользовательский интерфейс модуля виртуальной сборки кубика Рубика). В этом поможет библиотека JavaFX, которая отвечает за создание программного окна и отслеживание событий клавиатуры и мыши. В середине программного окна расположена модель кубика Рубика, которая может быть повернута с помощью движений мыши. По краям находятся элементы, при взаимодействии с которыми пользователь может вращать грани и задавать целый алгоритм для исполнения. Также расположены кнопки Scramble (размешивание кубика) и Solve (сборка из разобранного положения). В правой части экрана расположена панель для отображения алгоритма сборки головоломки.

Чтобы создать точку входа в JavaFX программу, нужно унаследовать класс от абстрактного класса Application и переопределить метод start, который и будет местом старта программы, после чего вызвать метод launch. Для создания виртуального пространства используется класс Scene, который хранит все 3D объекты и из координаты. Чтобы создать плавные анимации поворотов граней кубика, можно воспользоваться методами класса SequentialTransition и Timeline. Для корректного отображения 3D пространства используется класс PerspectiveCamera.

# 3 Анализ полученных результатов

В результате работы программы пользователь получает полный алгоритм (текстовую формулу) из любого заданного начального положения. Она представляет из себя ряд движений, в среднем равный 140-160 ходам. Пользователь может проделать все те же действия на реальной физической модели кубика Рубика, чтобы более эффективно обучаться. Средний алгоритм, составленный данной программой, займет 40-50 секунд на выполнение, что является приемлемым результатом для начинающего свой путь спидкубера.

Программа будет иметь неоптимальный результат в том случае, когда кубик размешан незначительно (см. Рис. 3 Пример незначительного размешивания головоломки). Полученный алгоритм будет такой же длины, как и при полностью запутанной головоломке. Дело в том, что программа составляет алгоритм вне зависимости от начального положения. В этом случае будет получен алгоритм в 140-160 ходов, а не 1. Такие ситуации являются специфическими, на остальных вариантах начального положения программа работает корректно.

# Заключение

В ходе данной работы был разработан модуль виртуальной сборки кубика Рубика, позволяющий пользователю научиться собирать данную головоломку и визуализировать этот процесс на 3D модели. Алгоритм, полученный в результате работы программы, - строка из специальных символов языка вращений. Данный алгоритм разделен на этапы, которые помогут структурировать действия при сборке.

В качестве языка программирования был выбран кроссплатформенный язык Java. Для работы с 3D графикой и событиями клавиатуры и мыши была задействована библиотека JavaFX.

При анализе результатов было выявлено, что средний алгоритм сборки составляет порядка 140-160 действий и занимает примерно 40-50 секунд. В будущем планируется добавить более эффективные методики сборки для более продвинутых пользователей.

# 

# Список использованных источников

1. Герберт Шилдт. Java. Полное руководство, 10-е издание = Java. The Complete Reference, 10th Edition. — М.: [«Диалектика»](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2018. — 1488 с.
2. Брюс Эккель. Философия Java = Thinking in Java. — 4-е изд. — СПб.: [Питер](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2018. — 1168 с.
3. А. Карасев. [Как научиться собирать кубик Рубика в объёме](http://www.nkj.ru/articles/228/9223) // [Наука и жизнь](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0_%D0%B8_%D0%B6%D0%B8%D0%B7%D0%BD%D1%8C). — 1999. — № 5.
4. Бенджамин Дж. Эванс, Джеймс Гоф, Крис Ньюленд. Java: оптимизация программ. Практические методы повышения производительности приложений в JVM. — М.: [Диалектика](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2019. — 448 с.
5. Кей С. Хорстманн. Java. Библиотека профессионала, том 1. Основы. 10-е издание = Core Java. Volume I - Fundamentals (Tenth Edition). — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2017. — 864 с.
6. И. Константинов. [Векторное сложение кубика](http://www.nkj.ru/articles/228/9222) // [Наука и жизнь](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0_%D0%B8_%D0%B6%D0%B8%D0%B7%D0%BD%D1%8C). — 1999. — № 5.

# Приложение

# Рисунок 1. Этапы сборки кубика Рубика.

# 

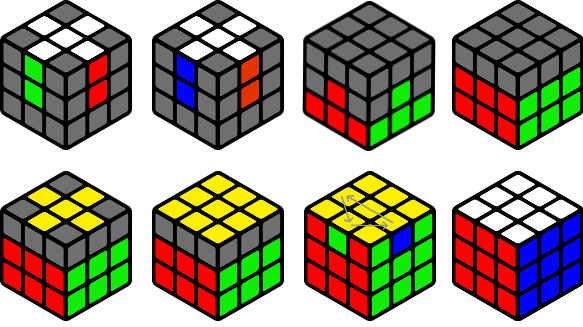


Рисунок 2. Пользовательский интерфейс модуля виртуальной сборки кубика Рубика.

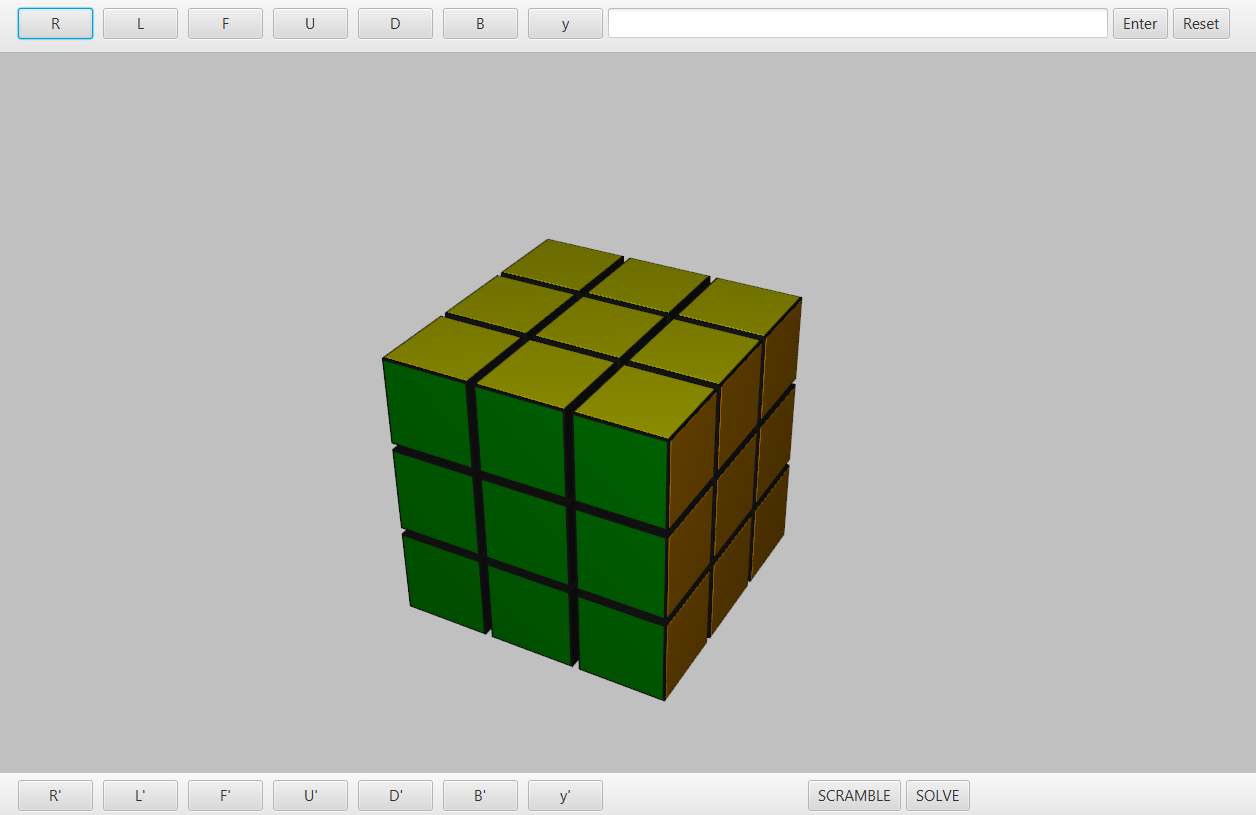


Рисунок 3. Пример незначительного размешивания головоломки.

