

УДК 004.514.6

## РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА НА ЯЗЫКЕ C++ ДЛЯ СОЗДАНИЯ, РЕДАКТИРОВАНИЯ И СОХРАНЕНИЯ ДРАКОН-СХЕМ

*Бурцева С.Н.*

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина, г. Рязань, Российская Федерация*

*Научный руководитель: Бакулева М.А. - к.т.н., доцент, доцент кафедры САПР ВС*

**Аннотация:** В статье рассматривается идея построения ДРАКОН-схем и принципы разработки графического редактора на C++ QT.

**Ключевые слова:** ДРАКОН-схемы, библиотека QT, графический редактор

**Введение.** Язык ДРАКОН (DRAKON) был создан при участии Федерального космического агентства России (НПЦ АП Н.А. Пилюгина, Москва) и Российской академии наук (ИПМ им. М.В. Келдыша, Москва). На основе ДРАКОНа была разработана автоматизированная технология проектирования алгоритмов и программ под названием "ГРАФИТФЛОКС", которая успешно применяется во многих крупных проектах, включая "Ангара", "Фрегат", "Морской старт" и другие.

Визуальный язык ДРАКОН обладает уникальными характеристиками в эргономике. В отличие от обычных блок-схем, ДРАКОН-схемы сохраняют ясность даже при сложных алгоритмах, а новый способ изображения логических операций при помощи инфографики делает их более прозрачными и удобными для понимания [1, 2].

**Основная часть.** ДРАКОН-схемы представляют собой графический язык программирования, предназначенный для проектирования и визуализации алгоритмов.

Существует несколько причин, почему ДРАКОН-схемы могут быть полезны:

1 **Наглядность:** ДРАКОН-схемы представляют алгоритмы в виде графических элементов, что обеспечивает высокую степень наглядности. Это упрощает понимание структуры программы и последовательности её выполнения.

2 **Простота использования:** Создание ДРАКОН-схем не требует специальных навыков программирования. Этот графический язык достаточно интуитивно понятен и может быть использован как программистами, так и людьми без опыта в программировании.

3 **Анализ и отладка:** ДРАКОН-схемы могут использоваться для анализа алгоритмов и выявления потенциальных проблем до начала программирования. Они помогают выявить логические ошибки и неясные моменты в алгоритме.

4 **Обучение программированию:** ДРАКОН-схемы могут быть полезны для обучения основам программирования. Они помогают начинающим программистам лучше понять основные концепции, такие как последовательность, условия и циклы.

5 **Документация:** ДРАКОН-схемы могут служить в качестве документации к программному коду. Они помогают другим разработчикам лучше понять структуру программы и её логику.

6 **Планирование проекта:** ДРАКОН-схемы могут использоваться для планирования и проектирования программных проектов. Они помогают разработчикам лучше представить структуру программы и определить последовательность выполнения задач.

**Дракон-схемы и их составляющие.** Дракон-схемы представляют собой совокупность действий. Под действием понимается нечто, что имеет конечную продолжительность и приводит к желаемому и совершенно определённом результату.

Использование ДРАКОН-схем может значительно улучшить понимание и эффективность разработки программных проектов, особенно на ранних стадиях проектирования.

Пример ДРАКОН-схемы из научных публикаций (рисунок 1):



Рисунок 1 – Пример ДРАКОН-схемы «История с попугаем»

Пример ДРАКОН-схемы в разработанном графическом редакторе (рисунок 2):

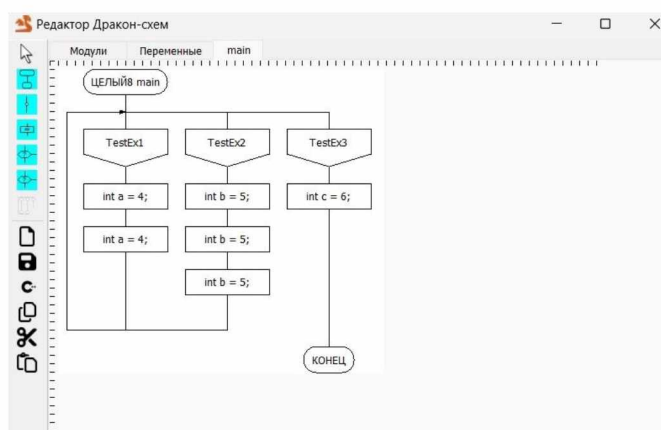


Рисунок 2 – Пример Дракон-схемы

QT и причины использования данной библиотеки для разработки графического редактора. В Qt C++ удобно создавать графический интерфейс для рисования ДРАКОН-схем по нескольким причинам:

– графические элементы: Qt предоставляет широкий набор графических элементов, таких как QGraphicsView, QGraphicsScene, QGraphicsItem и другие, которые могут быть использованы для создания интерактивного рисования ДРАКОН-схем.

– гибкость и настраиваемость: Qt позволяет создавать пользовательские элементы управления и настраивать их внешний вид и поведение в соответствии с вашими потребностями. Это позволяет создавать удобные инструменты для рисования ДРАКОН-схем с учетом специфических требований пользователя.

– событийная модель: Qt предоставляет мощную событийную модель, которая позволяет легко обрабатывать события мыши, клавиатуры и другие пользовательские действия. Это позволяет создавать интерактивные редакторы ДРАКОН-схем с поддержкой функций, таких как выделение, перемещение и масштабирование элементов.

– многоплатформенность: Приложения, разработанные с использованием Qt, могут быть легко импортированы на различные операционные системы, включая Windows,

macOS и Linux. Это обеспечивает удобство использования и доступность редактора ДРАКОН-схем для широкого круга пользователей [3, 4].

Отрисовка компонентов ДРАКОН-схемы. Можно рассмотреть отрисовку компонентов на примере метода `drawRoundedRect ()`.

Метод `drawRoundedRect ()` в Qt C++ используется для отрисовки прямоугольника с закругленными углами. Его сигнатура:

```
void QPainter::drawRoundedRect(const QRectF &rect, qreal xRadius, qreal yRadius, Qt::SizeMode mode = Qt::AbsoluteSize)
```

Этот метод принимает следующие параметры:

- `rect`: прямоугольник (QRectF), который нужно нарисовать.
- `xRadius`: радиус закругления по горизонтали.
- `yRadius`: радиус закругления по вертикали.

- `mode` (необязательный): режим размеров закругления. По умолчанию используется абсолютный размер (Qt::AbsoluteSize), но также можно использовать относительные размеры с помощью Qt::RelativeSize.

Пример кода, используемый в коде при написании графического редактора для отрисовки компонента “заголовок” ДРАКОН-схемы:

```
//Заголовок
qp.drawRoundedRect(headX, headY, TITLE_LEN, 2*ROUND_D, ROUND_D, ROUND_D, Qt::AbsoluteSize);
QFontMetrics fm(qp.font());
QString txt;
txt = type + “ “ + name;
QRectF a = fm.boundingRect(txt);
While(a.width() > TITLE_LEN){
    txt = txt.left(txt.length() - 1);
    a = fm.boundingRect(txt + “...”);}
qp.drawText(QRect(headX, headY, TITLE_LEN, 2*ROUND_D), QT::AlignCenter, txt);
```

**Заключение.** Таким образом язык ДРАКОН обладает уникальными характеристиками. ДРАКОН-схемы - графический язык программирования, предназначенный для проектирования и визуализации алгоритмов.

### Список литературы

1. Алгоритмы и жизнеритмы на языке ДРАКОН. Разработка алгоритмов. Безошибочные алгоритмы. — М., 2019. — 374 с. — Иллюстраций: 195.
2. Ковальчук М., Михайленко Л. Алгоритмическая культура как компонент алгоритмической деятельности // Knowledge, Education, Law, Management 2018 № 1 (21)
3. Филатова Л.Ю., Филатова А.С. Развитие алгоритмического стиля мышления при обучении студентов вуза // Наука ЮУрГУ: материалы 67-й научной конференции. Секции естественных наук. — С. 469-472.
4. Грис Д. Наука программирования. — М.: Мир, 1984. — С. 303.

UDC 004.514.6

## DEVELOPMENT OF A GRAPHIC EDITOR IN C++ FOR CREATING, EDITING AND SAVING DRAGON DIAGRAMS

*Burtseva S.N.*

*Ryazan State Radio Engineering University named after V.F. Utkin, Ryazan, Russian Federation*

*Bakuleva M.A. - Cand. of Sci., associate professor, associate professor of the department of CAD CS*

**Abstract:** The article discusses the idea of building DRAGON diagrams and the principles of developing a graphical editor in C++ QT.

**Keywords:** DRAGON diagrams, QT library, graphic editor