**Слайд 1:**

Здравствуйте, уважаемый председатель и члены государственной комиссии. Вашему вниманию представляется работа, посвященная разработке и проектированию транслятора с языка КуМир в язык программирования С++.

**Слайд 2.**

Целью данной работы является разработка транслятора со школьного алгоритмического языка программирования КуМир в язык программирования C++ для снижения входного порога в область робототехники как со стороны ученика, предоставляя возможность заниматься разработкой роботов, используя полученные навыки программирования в системе КуМир, так и со стороны преподавателя, уменьшая затраты на приобретение программных и аппаратных средств разработки.

**Слайд 3.**

Среда программирования КуМир – разработка с открытым исходным кодом. Для данного проекта имеется поддержка дополнений и доработок. Одна из таких доработок представлена в данной работе. Доработка заключается в добавлении возможности программирования внешних устройств, разработанных на базе плат Arduino.

**Слайд 4.**

На слайде приведен план работы. Первый этап работы заключается в поиске и изучении необходимой информации для выполнения поставленной цели. Второй блок заключается в анализе исходных кодов среды программирования Кумир и разработке транслятора с языка КуМир в язык С++ и исполнителя Arduino для предоставления набора основных команд этой платформы.

**Слайд 5.**

Итак, данная работа посвящена разработке и проектированию транслятора с языка КуМир в язык программирования С++. Анализ теоретических источников по теме исследования показал, что спектр возможностей применения системы программирования КуМир ограничен, ряд стандартных команд позволяет разрабатывать небольшие приложения с целью изучения основ программирования на процедурных языках.

**Слайд 6.**

На слайде приведена архитектура среды программирования КуМир.

Система КуМир состоит из расширяемого набора исполнителей, различных систем программирования и вспомогательных утилит. В ходе работы был разработан плагин для трансляции исходного кода в язык программирования С++, а также исполнитель Arduino (выделены зеленым цветом на слайде).

**Слайд 7.**

На слайде приведен полный цикл разработки робота на языке программирования КуМир: исходный код транслируется в код на языке С++, после чего происходит прошивка платы.

**Слайд 8.**

Основными используемыми объектами при трансляции являются плагины - утилиты, предоставляющие интерфейс для выполнения различных действий - компиляции и анализа исходного кода, генерации файла с исполняемым кодом, а также исполнения транслированного кода на виртуальной низкоуровневой машине.

**Слайд 9.**

За хранение информации о AST-дереве разбора отвечает соответствующая структура, содержащая определения для различных выражений, модулей и утверждений, типов, а также алгоритмов.

**Слайд 10.**

модуль - комплекс программного кода.

выражения - переменные, константы, вызовы функций, а также вложенные выражения.

утверждения - циклы, управляющие метки, условные выражения, команды взаимодействия с пользовательским вводом.

тип описывает абстрактный тип данных

алгоритм описывает информацию о функциях и программе в целом.

**Слайд 11.**

Для реализации исполнителей на языке C++ в системе Кумир 2.0 был создан инструмент, позволяющий создать единый шаблон разработки исполнителей, используя декларативное описание. Под ним подразумевается файл в формате JSON, который является частью проекта. Каждый исполнитель представляется отдельным проектом при компиляции исходных кодов.

**Слайд 12.**

Среда программирования КуМир – проект-долгожитель. Первый коммит в репозитории среды датируется 22 апрелем 2011 года. За время разработки объем кодовой базы проекта рос, разработка велась разными людьми и командами. В процессе анализа и взаимодействия с исходными кодами, был выявлен ряд сложностей, повлиявших на трудоемкость работы.

* Ни одна из современных сред разработки для работы с используемыми в проекте библиотеками не способна скомпилировать проект с использованием как 4 версии фреймворка QT, так и 5
* отладка проекта в момент исполнения в привычном понимании недоступна - возможность отслеживания изменений при помощи точек останова отсутствует.
* отсутствует документация к исходным кодам системы программирования КуМир, правила оформления программного кода, а объем комментариев не покрывает и доли процента программного кода проекта.

**Слайд 13.**

Для трансляции программного кода с исходного языка программирования в выходной язык был изменен набор используемых инструкций. Инструкции разделяются на 3 группы – управляющие команды, инструкции-разделители и инструкции-операции.

Основные управляющие инструкции приведены на слайде.

**Слайд 14.**

На следующем слайде приведены инструкции-разделители. Далее расположены инструкции-операции.

**Слайд 15.**

В целом, блоки трансляции различных инструкций схожи между собой. Наиболее интересными моментами являются – трансляция циклов, подвыражений и переменных.

**Слайд 16.**

Существует 4 типа циклов в языке программирования КуМир:

* бесконечный цикл
* цикл n раз
* цикл со счетчиком
* цикл с предусловием

**Слайд 17.**

При трансляции в низкоуровневый язык, этап трансляции циклов требовал одной инструкции -``LOOP``, а также операций, выполняющихся до данной метки. В начале транслировалось тело цикла, в случае цикла с предусловием до тела транслировалось условие исполнения, на последнем этапе трансляции добавлялась инструкция ``LOOP``, объявляющую метку завершения определения цикла.

**Слайд 18.**

Для трансляции циклов в язык С++ были добавлены инструкции для соответствующих типов циклов --- ForLoop, WhileLoop.

**Слайд 19.**

В начале трансляции цикла любого типа указывается заголовок, определяющий тип цикла и возможные предусловия. В случае цикла ``нц пока``, заголовок содержит условие окончания работы, состоящее из ряда выражений.

**Слайд 20.**

На слайде приведен пример подвыражения и его структурное представление в ходе выполнения программы.

Любое подвыражение представляется бинарным деревом, в вершине которого находится операнд, а на листьях - константы или переменные. Для корректной трансляции подвыражений требуется разобрать дерево подвыражения сверху вниз - найти узел, листья которого не содержат дальнейшей вложенности, затем добавить в список инструкций левый лист узла, затем сам узел, в конце - правый лист.

Процесс трансляции подвыражений реализован в методе calculate.

**Слайд 21.**

Он используется во множестве мест и содержит логику для трансляции переменных, констант, вызовов функций и подвыражений. Поскольку метод является рекурсивным, невозможно добавить дополнительный блок операций трансляций, описанный выше, в тело метода calculate, поэтому было принято решение вынести логику данного метода в новый - innerCalculation, превратив исходный метод в обертку, куда и была добавлена вспомогательная логика.

**Слайд 22.**

На слайде приведен пример трансляции нескольких переменных.

В разработанной реализации в случае использования переменной в список инструкций добавляется инструкция ``VAR``, хранящая ссылку на название переменной и на тип, в случае объявления.

**Слайд 23.**

На данный момент планируется расширить возможности разработанного транслятора до отдельного приложения для персональных компьютеров, а также добавить

* возможность прошивки робота из клиента
* инструмент выбора порта с подключенным роботом
* настраиваемый алгоритмом прошивки.

**Слайд 24.**

В ходе работы были изучены и проанализированы исходные кода среды программирования КуМир, был разработан транслятор с языка программирования КуМир в язык C++, а также модуль для взаимодействия с программной платформой Arduino при помощи языка программирования КуМир, а также решен ряд поставленных задач