Презентация «KAA-2020» (примерное время показа: 10-15 минут)

Слайд 1

**«KAA-2020»**

Ковалев Александр

ПОИТ-4

2 курс

Слайд 2

Язык программирования KAA-2020 - это компилируемый язык программирования высокого уровня со статической типизацией, который поддерживает следующие парадигмы программирования: процедурное, модульное, структурное программирование.

Это примитивный язык программирования, который может быть использован в следующих целях:

1. Как инструмент обучения построению простых алгоритмов;
2. Для выполнения простых математических вычислений.

В языке присутствуют 5 фундаментальных типов: строковый string, целочисленный со знаком int, целочисленный беззнаковый uint, символьный char и логический bool. Входной точкой программы является функция main.

Слайд 3

Помимо функционала, представленного в качестве задания к курсовому проекту, в языке KAA-2020 присутствуют так же следующие особенности:

1. Возможность включения подсветки синтаксиса в приложении Notepad++ путём подключения файла, содержащего иерархическую структуру тегов XML, описывающих особенности отображения конструкций языка;
2. Управление выводом отладочной информации транслятора KAA-2020 при помощи трёх ключей;
3. Наличие манипуляторов вывода, управляющих выводом целочисленных значений и позволяющих выводить определённые символы в консоль;
4. Возможность корректного выполнения вложенных циклов с контролем области видимости переменных, объявленных вне их и внутри;
5. Наличие одной директивы препроцессора, позволяющей поместить в исходный код содержимое другого текстового файла;
6. Трансляция на язык ассемблера;
7. Наличие двух операторов преобразования типа.

Слайд 4

Контрольный пример на данном слайде позволяет более-менее полноценно продемонстрировать отдельные особенности языка KAA-2020. В нём присутствуют:

1. Использование директивы import;
2. Вывод строк и целочисленных значений с использованием манипуляторов вывода;
3. Использование двух операторов примитивного преобразования типов;
4. Несколько целочисленных выражений со всеми шестью представленными в языке арифметическими операциями;
5. Трёхуровневый вложенный цикл.

Пройдёмся по некоторым из особенностей более подробно.

Слайд 5

«При реализации идеи о введении в мой язык манипуляторов вывода возникла проблема плохой читабельности кода. При этом, для создания более компактного кода, умещая несколько операторов в одну строку, было бы не плохо визуально отделить каждый из операторов. Таким образом я пришёл к следующему решению».

При написании кода на языке KAA-2020 в приложении Notepad++ присутствует возможность включения подсветки синтаксиса языка.

Это осуществляется путём перехода по вкладке «Синтаксисы», пункт «Задать свой синтаксис». В открытом окне необходимо выбрать кнопку «Импортировать». Далее, необходимо выбрать файл «KAA-2020\_sint.xml», который является дополнением к основному приложению транслятора. В данном файле представлена иерархическая структура тегов XML, описывающая форматирование отдельных конструкций исходного кода языка.

При этом выделено несколько групп ключевых слов:

1. Типы данных подсвечиваются голубым цветом, при этом шрифт полужирный;
2. Литералы true и false имеют пурпурный цвет;
3. Операторы for и out подсвечиваются зелёным цветом при этом шрифт полужирный;
4. Строковые литералы имеют бордовый цвет, а целочисленные – зелёным с наклонным шрифтом;
5. Манипулятора вывода имеют наклонный шрифт и подсвечиваются ярко голубым цветом и т.д.

Слайд 6

Транслятор языка KAA-2020 поддерживает приём в качестве параметров трёх ключей, влияющих на объём отладочной информации, выводимой на консоль и в текстовые файлы:

**/SIN** – вывод трассировки синтаксического разбора и правил в консоль (по умолчанию – только в протокол);

**/PN** – вывод в два текстовых файла результата работы преобразователя выражений в польскую нотацию;

**/СT** – вывод в консоль времени выполнения отдельных стадий работы транслятора.

Слайд 7

В результате использования ключа **/SIN** в консоль была выведена трассировка синтаксического разбора и правила, применяемые при синтаксическом анализе данного исходного кода.

Слайд 8

В результате использования ключа **/PN** объём отладочной информации, выводимой в консоль, остался неизменным, но в директории с проектом появилось два текстовых файла.

Слайд 9

Первый файл содержит таблицу лексем с выражениями, представленными в польской нотации. При этом символы операций ввиду особенностей реализации работы транслятора и для упрощения правил синтаксиса языка заменены на символ «v».

Во втором файле содержится таблица лексем с выражениями, представленными в польской нотации, при этом символы операций представлены в необходимом виде.

Слайд 10

В результате использования ключа **/CT** в консоль выводится информация о времени работы каждой стадии транслятора в виде количества затраченных миллисекунд, а также итоговое время работы всего транслятора в целом.

Слайд 11

Язык поддерживает вывод значений строковых и целочисленных литералов в соответствии с используемыми манипуляторами вывода. Всего в языке присутствует 7 манипуляторов вывода:

1. Манипулятор |e позволяет поставить символ перевода на строку;
2. Манипулятор |d устанавливает режим вывода целочисленных значений в десятичной системе счисления;
3. Манипулятор |t позволяет поставить символ перевода табуляции;
4. Манипулятор |b устанавливает режим вывода целочисленных значений в двоичной системе счисления;
5. Манипулятор |h устанавливает режим вывода целочисленных значений в шестнадцатеричной системе счисления;
6. Манипулятор |f выводит разделитель, позволяющий визуально ограничить информацию, выводимую на консоль
7. Манипулятор |o устанавливает режим вывода целочисленных значений в восьмеричной системе счисления.

Слайд 12

На данном и следующем слайдах представлены результаты использования манипуляторов вывода. Вывод значения переменной «b» в четырёх системах счисления и вывод разделителя, отделяющая текст, выводимый на консоль.

Слайд 13

Этот слайд демонстрирует работу вложенных циклов и манипуляторов вывода, устанавливающих символы перевода на строку и табуляции.

Слайд 14

В языке присутствует возможность использования директивы препроцессора **import**, позволяющая ещё до стадии лексического анализа «вставить» в исходных код программы текст из другого файла с кодом.

Синтаксис данной директивы представлен на слайде: после директивы import указывается имя текстового файла в треугольных скобках. При этом следует отметить, что файл, указываемый при использовании директивы, должен быть помещён в папку с именем «Library», находящуюся в директории с приложением транслятора.

На слайде представлен пример использования директивы, а так же содержимое файла, подключаемого к файлу с исходным кодом.

Слайд 15

Результат работы данной директивы можно увидеть при помощи файла протокола работы транслятора.В итоге содержимое файла «math.txt» было помещено в начало файла с исходным кодом. При этом директива import может быть употреблена в любом месте текста исходного кода.

Слайд 16

Язык KAA-2020 поддерживает два оператора преобразования типов:

1. (uint) <int> - реализует преобразование целочисленного значения типа int в беззнаковое число. При этом знаковый бит числа, хранимого в двоичном виде, будет интерпретироваться как значимый.
2. (int) <uint> - реализует преобразование беззнакового целочисленного значения типа uint в число со знаком. При этом последний бит числа, хранимого в двоичном виде, будет интерпретироваться как знаковый.

Слайд 17

На данном слайде представлен результат работы двух операторов преобразования типов. Подобный результат работы оператора (uint) можно объяснить тем, что последний бит числа переменной «a», установленный в единицу, начинает интерпретироваться как значимый, тем самым в разы увеличивая преобразуемое значение.

Слайд 18

**Спасибо за внимание!**