

Вариант 79 (***)

Разработать систему для управления клеточным роботом, осуществляющим передвижение по клеточному лабиринту. Клетка лабиринта имеет форму квадрата, клетки лабиринта могут быть разделены стенами (толщиной стен можно пренебречь), из различных материалов (пластик, дерево, бетон, и т.п.)

Робот может передвинуться в соседнюю клетку в случае отсутствия между ними препятствия, у робота есть ограниченные возможности разрушать препятствия (см. соответствующий раздел).

1. Разработать формальный язык для описания действий клеточного робота с поддержкой следующих литералов, операторов и предложений:

- Знаковых целочисленных литералов в десятичном формате;
- Логические литералы **false**, **true**, **undefined**;
- Строковые литералы задаются в двойных (“строка”) или одинарных (‘строка’) кавычках
- Объявление переменных и констант в форматах:
 - Логическая переменная **bool** <имя переменной1>[= логическое выражение][, <имя переменной2>[= логическое выражение], ...]; значения true и false;
 - Целочисленная переменная **int** <имя переменной1>[= арифметическое выражение][, <имя переменной2>[= арифметическое выражение], ...];
 - Строковая переменная **str**<имя переменной1>[= строковое выражение][, <имя переменной2>[= строковое выражение], ...];
 - Объявление одномерных массивов **array of** <тип элемента> <имя переменной1>, [<имя переменной2> ...]; массив динамический, при необходимости может увеличиваться (или уменьшаться) в размерах;
- Доступ к элементу массива <имя переменной> [индекс]; индексация элементов с 0; возвращает ссылку на объект.
- Операторы преобразования типа
 - (<имя типа >) <выражение>
 - преобразования в строку определены для всех типов (включая массив); при преобразовании в вектор атомарного типа получается вектор с элементами заданного типа.

Применяется строгая типизация, если типы не совпадают, то это семантическая ошибка.

- Оператор присваивания:
 - <переменная> = <выражение> присвоение левому операнду значения правого; оператор право ассоциативен;
- Арифметических / логических / строковых / поэлементных бинарных операторов сложения (для строк конкатенации; для логических “или”), вычитания (для строк – разность; для логических типов “или-не”); операторы возвращают временный объект с результатом вычислений:
 - <выражение> +|- <выражение>
 - размерность и типы массивов должны совпадать
- Операторов сравнения (<, >, =, !=), возвращают true при выполнении условия, false при не выполнении; сравнения для массивов выполняются, начиная с первого элемента:
 - < выражение> оператор < выражение>;
- Объединение предложений в группы с помощью оператора **start** и **stop**;
- Операторов цикла **enter** <логическое выражение> **do** <предложение языка / группа предложений> **until** <логическое выражение>, тело цикла выполняется до тех пор, пока выражение в условии является истинным, цикл начинается, только если условие в блоке **enter** является истинным, далее это условие не проверяется.
- Операторов управления роботом
 - перемещения робота на одну клетку вправо (**right**), влево (**left**), вперед (**forward**),

- назад (**back**); робот может перемещаться не поворачиваясь; если оператор невозможно выполнить из-за наличия препятствия, оператор вернет true, иначе false.
- поворот робота направо (**rotate_right**), налево (**rotate_left**); всегда возвращает true;
- измерения расстояния до ближайшего препятствия с помощью точного лазерного дальномера и рефлектометра (**measure**), возвращается расстояние до препятствия, а в связанной с рефлектометром переменной появляется тип материала;
- связь переменной с рефлектометром **bind** <имя переменной>
- описание материала препятствия задается в виде массива сток с описанием материала (Пр.: “WOOD”, “STEEL”, “CONCRETE”, “PLASTIC”); выход из лабиринта обозначается строкой “EXIT”; материалы обладают определенной прочностью (задается средой выполнения); стены могут быть из комбинации материалов (Пр: “STEEL, GLASS, PLASTIC, GLASS”) с прочностью пропорциональной прочности компонент.
- уничтожить препятствие по направлению движения при помощи бура (**demolish**); прочность бура конечна (задается средой выполнения) и уменьшается на прочность разрушенного препятствия; возвращает уровень оставшейся прочности; если прочности не хватает, то препятствие остается, а прочность бура уменьшается до нуля.
- Описатель функции
 - <тип возвращаемого значения> **function** <имя функции> ([<имя параметра1> = <значение по умолчанию 1>],...) <предложение языка / группа предложений языка>
 - Функция является отдельной областью видимости, параметры передаются в функцию по значению; из функции параметр возвращается по значению.
 - Возвращаемое значение является значением после оператора **return** <выражение>.
 - Точкой входа в программу определяется оператором **begin**(<имя переменной с именем функции> | <имя функции>);
 - Функция может быть определена внутри другой функции; в этом случае она имеет доступ ко всем переменным родительской функции определенным до ее объявления;
 - Функция может принимать в качестве параметра имя другой функции
- Оператор вызова функции
 - <имя функции> ([<имя параметра 1>,...]), вызов функции может быть в любом месте программы;
 - **call** (<имя строковой переменной с именем функции>[, (<имя строкового массива с именами параметров>)]

Предложение языка завершается символом ‘;’. Язык является регистрозависимым.

2. Разработать с помощью flex и bison интерпретатор разработанного языка. При работе интерпретатора следует обеспечить контроль корректности применения языковых конструкций (например, инкремент/декремент константы); грамматика языка должна быть по возможности однозначной.

3. На разработанном формальном языке написать программу для поиска роботом выхода из лабиринта. Описание лабиринта, координаты выхода из лабиринта и начальное положение робота задается в текстовом файле, выход из лабиринта замурован в стене лабиринта, робот может обнаружить выход столкнувшись со стеной либо при помощи дальномера.