

Интерпретатор для клеточного робота

Вариант 60 (***)

Разработать систему для управления клеточным роботом, осуществляющим передвижение по клеточному лабиринту. Клетка лабиринта имеет форму квадрата.

Робот может передвинуться в соседнюю клетку в случае отсутствия в ней препятствия.

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

- Знаковых целочисленных литералов в десятичном формате;
- Логических литералов **TRUE** и **FALSE**; логические константы и выражения преобразуются к знаковым целочисленным как 1 и 0 соответственно, обратное преобразование не определено;
- Объявление переменных в форматах:
 - Целочисленная переменная со знаком **INT** **<имя переменной> = <арифметическое выражение>;**
 - Логическая переменная **BOOLEAN** **<имя переменной> = <логическое выражение>;**
 - Карта мира **MAP** **<имя переменной>;**
- Объявление констант в форматах:
 - Целочисленная константа со знаком **CINT** **<имя переменной> = <арифметическое выражение>;**
 - Логическая константа **CBOOLEAN** **<имя переменной> = <логическое выражение>;**

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

- Операторов присваивания **:=**;
 - Арифметических операторов (результат инкремент/декремент первого операнда и возврат соответствующего значения):
 - Инкремент значения целочисленной переменной **INC** **<имя переменной | арифметическое выражение | вызов процедуры> <имя переменной | арифметическое выражение | вызов процедуры | целочисленный литерал | логическое выражение | TRUE | FALSE>**
 - Декремент значения целочисленной переменной **DEC**, аналогично **INC**.
 - Логических операторов (результат логическое выражение):
 - **NOT** **<логическое выражение | вызов процедуры > ;**
 - **OR** **<логическое выражение | вызов процедуры | TRUE | FALSE > <логическое выражение | вызов процедуры | TRUE | FALSE >;**
 - Операторов сравнения:
 - **GT** **<арифметическое выражение> <арифметическое выражение>;**
 - **LT** **<арифметическое выражение> <арифметическое выражение>;**
- (все операторы для записи выражений равноприоритетны; операторные скобки не используются; см. обратная польская запись).

- Операторов цикла **WHILE** <логическое выражение> **DO** <предложение языка / группа предложений>
- объединение предложений в группы с помощью скобок ();
- Условных операторов **IF** <логическое выражение> <предложение языка / группа предложений> с поддержкой **ELSE** <предложение языка / группа предложений>;
- Операторов управления роботом
 - перемещения робота на одну клетку в заданном направлении относительно текущего **STEP**. Если оператор невозможно выполнить из-за наличия препятствия, он возвращает логическое значение **FALSE**. Иначе – **TRUE**.
 - Поворот робота **BACK, RIGHT, LEFT**. После выполнения оператора робот разворачивается в указанном направлении относительно текущего/
 - Измерение расстояния до первого препятствия **LOOK**, возвращает количество клеток до ближайшего препятствия в текущем направлении.
- Описатель процедуры
 - **PROC** имя процедуры [имена переменных разделенных пробелом] группа предложений языка, типы переменных выводятся из их объявления в теле процедуры. Процедура является отдельной областью видимости, параметры передаются в функцию по ссылке, т.е. через них происходит также возврат результата. Процедура может быть объявлена в любом месте программы, при объявлении она не выполняется.
- Оператор вызова процедуры
 - **имя процедуры** [имена переменных разделенных пробелом], вызов процедуры может быть в любом месте программы после ее объявления, при вызове процедуры в выражениях в качестве результата используется первый из параметров подходящего типа, неявное преобразование типов в этом случае не производится.
- Для работы с картами используются операторы с форматом вызова в виде вызова процедуры с четырьмя параметрами: переменная результата, имя карты, координата по X, координата по Y:
 - **BAR** – проверить наличие препятствия в клетке;
 - **EMP** – проверить отсутствие препятствия в клетке;
 - **SET** – установить препятствие в клетку (1) или определить, что ячейка свободна;
 - **CLR** – сбросить ячейку в неопределенное состояние;

Предложение языка завершается символом перевода строки. Язык является регистронезависимым, т.е. $INT = int = InT = \dots$

2. Разработать с помощью flex и yacc 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 и начальное положение робота задается в текстовом файле.

