

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы трансляции

ОТЧЁТ
по лабораторной работе
на тему

Определение модели языка. Выбор инструментальной языковой среды

Выполнил
Студент гр. 053501
Хиль В.М.

Проверил
Ассистент кафедры информатики
Гриценко Н.Ю.

Минск 2023

СОДЕРЖАНИЕ

1	Цель работы	3
2	Подмножество языка программирования	4
2.1	Константы(литералы)	4
2.2	Типы данных.....	4
2.3	Выражения	5
2.4	Операторы цикла.....	5
2.5	Условные операторы.....	6
3	Инструментальная языковая среда.....	7
	Приложение А (обязательное) Нахождение факториала	8
	Приложение Б (обязательное) Нахождение чисел Фибоначчи	9

1 ЦЕЛЬ РАБОТЫ

Необходимо определить подмножество языка программирования (типы констант, переменных, операторов и функций). В подмножество как минимум должны быть включены:

- числовые и текстовые константы;
- 3-4 типа переменных;
- операторы цикла (while, for);
- условные операторы (if...else).

Определение инструментальной языковой среды, т.е. языка программирования и операционной системы для разработки включает:

- язык программирования с указанием версии, на котором ведётся разработка (напр. Python 3.7);
- операционная система (Windows, Linux и т.д.), в которой выполняется разработка;
- компьютер (PC / Macintosh).

В отчете по лабораторной работе дается полное определение подмножества языка программирования, тексты 2-3-х программ, включающих все элементы этого подмножества. Приводится подробное описание инструментальной языковой среды.

2 ПОДМНОЖЕСТВО ЯЗЫКА ПРОГРАММИРОВАНИЯ

В качестве подмножества языка программирования выбран язык BASIC.

Язык создавался как инструмент, с помощью которого студенты-непрограммисты могли самостоятельно создавать компьютерные программы для решения своих задач. Получил широкое распространение в виде различных диалектов, прежде всего как язык для домашних компьютеров. К настоящему моменту претерпел существенные изменения, значительно отойдя от характерной для первых версий простоты, граничащей с примитивизмом, и превратившись в достаточно ординарный язык высокого уровня с типичным набором возможностей. Используется как самостоятельный язык для разработки прикладных программ, главным образом, работающих под управлением ОС Windows различных версий.

Существует огромное множество версий BASIC-а, но в данных лабораторных работах я буду рассматривать процедурную версию интерпретируемого BASIC-а с динамической типизацией.

2.1 Константы(литералы)

Поддерживаются следующие литералы:

- 1, -4, 0 (int);
- 141.324 (float);
- "Hello World!" (string).

2.2 Типы данных

Поддерживаются следующие типы данных:

- int: представляет целое число. В зависимости от архитектуры процессора может занимать 2 байта (16 бит) или 4 байта (32 бита). Диапазон предельных значений соответственно также может варьироваться от –32768 до 32767 (при 2 байтах) или от -2 147 483 648 до 2 147 483 647 (при 4 байтах);
- float: представляет вещественное число ординарной точности с плавающей точкой в диапазоне +/- 3.4E-38 до 3.4E+38. В памяти занимает 4 байта (32 бита);
- string: тип string предназначен для работы со строками.

Поддерживаются следующие неявные преобразования: из int float.

2.3 Выражения

Выражение – конструкция языка, возвращающее некоторое значение. Поддерживаются следующие выражения:

- Выражение литерала. Представляет собой один токен литерала. Возвращает значение, соответствующее данному литералу.

- Выражение идентификатора. Синтаксис: <переменная>. Возвращает значение переменной.

- Унарное выражение. Имеет следующий синтаксис: <унарный оператор><выражение>. Применяет унарную операцию к выражению. Поддерживаемые унарные операторы: + и - для числовых выражений.

- Бинарное выражение. Синтаксис: <выражение 1><бинарный оператор><выражение 2>. Применяет бинарную операцию к 2 выражениям. Типы выражений 1 и 2 должны быть эквивалентными. Возвращает выражение такого же типа. Поддерживаемые бинарные операторы +, -, *. Очередность, в которой выполняются операторы эквивалентна их очередности в языке BASIC.

- Выражение присваивания. Синтаксис: <переменная>=<выражение>. Возвращает тип, соответствующий типу <выражения>, одновременно записывая это значение в <переменную>. <выражение> вычисляется один раз.

- Выражение вызова. Синтаксис: <функция>(<список аргументов>). Возвращает значение функции. <список аргументов> представляет собой одно или несколько выражений, разделённых токенами запятых.

2.4 Операторы цикла

Поддерживаются следующие операторы цикла:

- Цикл while выполняет некоторый код, пока его условие истинно, то есть возвращает true. После ключевого слова while идет условное выражение, которое возвращает true или false. Затем идет набор инструкций, которые составляют тело цикла. И пока условие возвращает true, будут выполняться инструкции в теле цикла.

- Цикл for. В данном коде, инициализация – операция присваивания, в которой устанавливается начальное значение переменной цикла. Эта переменная является счетчиком, который управляет работой цикла. Выражение – условное выражение, в котором проверяется значение переменной цикла. На этом этапе определяется дальнейшее выполнение цикла. Прирост – определяет, как будет изменяться значение переменной цикла после каждой итерации. Цикл for выполняется до тех пор, пока значение выражения равно true. Как только значение выражения станет false, выполнение цикла прекращается и выполняется оператор, который следует за циклом for.

- `continue` – начинает следующий проход цикла, не исполняя оставшееся тело цикла.
- `break` – прерывает исполнение цикла.

2.5 Условные операторы

Поддерживаются следующие условные операторы:

- `if` – если условие истинно, то выполняется блок кода, который идет после `if`.
- Связка `if-elif-else` – если условие истинно, то выполняется блок кода, который идет после `if`, иначе выполняется блок кода, который идет после `elif`, иначе выполняется блок кода, который идет после `else`.
- Связка `if-else` – если условие истинно, то выполняется блок кода, который идет после `if`, иначе выполняется блок кода, который идет после `else`.

3 ИНСТРУМЕНТАЛЬНАЯ ЯЗЫКОВАЯ СРЕДА

В качестве языковой среды выбран язык программирования Python. Разработка основана на работе с операционной системой Windows на РС.

Python — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML). Разработчики используют Python, потому что он эффективен, прост в изучении и работает на разных платформах.

Язык Python имеет следующие преимущества:

- 1) Разработчики могут легко читать и понимать программы на Python, поскольку язык имеет базовый синтаксис, похожий на синтаксис английского.
- 2) Python помогает разработчикам быть более продуктивными, поскольку они могут писать программы на Python, используя меньше строк кода, чем в других языках.
- 3) Python имеет большую стандартную библиотеку, содержащую многократно используемые коды практически для любой задачи. В результате разработчикам не требуется писать код с нуля.
- 4) Активное сообщество Python состоит из миллионов поддерживающих разработчиков со всего мира. При возникновении проблем сообщество поможет в их решении.
- 5) Кроме того, в Интернете доступно множество полезных ресурсов для изучения Python. Например, можно легко найти видеоролики, учебные пособия, документацию и руководства для разработчиков.
- 6) Python можно переносить на различные операционные системы.

ПРИЛОЖЕНИЕ А
(обязательное)
Нахождение факториала

```
FUN fact(n)
  VAR result = 1
  IF n < 0 THEN
    PRINT("Error! Factorial of a negative number doesn't exist.")
  ELSE
    FOR i = 1 TO n+1 THEN
      VAR result = result * i
    END
  END
  RETURN result
END

VAR a = fact(5)
PRINT(a)
```


ПРИЛОЖЕНИЕ Б
(обязательное)
Нахождение чисел Фибоначчи

```
FUN fibonacci(n)
  VAR e11 = 0
  VAR e12 = 1
  VAR next = e11 + e12

  PRINT(e11)
  PRINT(e12)

  WHILE next <= n THEN
    PRINT(next)
    VAR e11 = e12
    VAR e12 = next
    VAR next = e11 + e12
  END
END

VAR a = fibonacci(100)
PRINT(a)
```