**Теоретические сведения к индивидуальным практическим работам**

Общие сведения о метриках сложности программ

Качество программных средств во многом зависит от сложности их кодов. Например, чем сложнее программа, тем ниже ее надежность и сопровождаемость. Поэтому при оценке качества программ обычно оценивается и их сложность.

Метрики сложности программ принято подразделять на ***три основные группы*** [2]:

* метрики размера программ;
* метрики сложности потока управления программ;
* метрики сложности потока данных программ.

**Метрики размера программ**

Метрики этой группы основаны на анализе исходных текстов программ.

Существуют различные метрики, с помощью которых может быть оценен размер программы.

К наиболее простым метрикам размера программы относятся ***количество строк исходного текста программы***и ***количество операторов программы***.

Из метрик размера программ широкое распространение получили ***метрики Холстеда*** [3].

Основу метрик Холстеда составляют *шесть базовых метрик* программы:

* 1 *–* словарь операторов (число уникальных операторов программы);
* 2 *–* словарь операндов (число уникальных операндов программы);
* *N1 –* общее число операторов в программе;
* *N2 –* общее число операндов в программе;
* *f1j –* число вхождений *j*-го оператора, *j* = 1, 2, …, 1;
* *f2i –* число вхождений *i*-го операнда, *i* = 1, 2, …, 2.

Справедливы следующие соотношения:





Базовые метрики определяются непосредственно при анализе исходных текстов программ. На основе базовых метрик Холстед предложил рассчитывать ряд производных метрик программы. Среди них рассмотрим следующие:

* словарь программы (общее число уникальных операторов и операндов программы):
* длина программы (общее количество операторов и операндов программы):
* объем программы (число битов, т.е. логических единиц информации, необходимых для записи программы):



Операнды программы представляют собой используемые в ней переменные и константы.

Под операторами программы в метриках Холстеда подразумеваются входящие в ее состав символы операций, символ присваивания, символы-разделители точка и точка с запятой, круглая скобка (пара из открывающей и закрывающей скобок считается одним оператором), управляющие операторы, составной оператор, а также имена процедур и функций.

Несколько служебных слов, входящих в состав одного оператора (например, If…Then…Else), считаются одним оператором.

Метки не относятся ни к операторам, ни к операндам.

Очевидно, что совокупность операторов программы и их количество зависят от языка программирования, на котором написана программа.

Операторы языка Паскаль в интерпретации Холстеда приведены в табл. 1. При подсчете количества операторов и операндов в программе, написанной на языке Паскаль, следует анализировать только ее раздел операторов, а также разделы операторов процедур и функций пользователя.

Таблица 1

**Операторы языка Паскаль в интерпретации Холстеда**

|  |  |
| --- | --- |
| Обозначение оператора | Назначение оператора |
| **+** | плюс (сложение, объединение множеств, сцепление строк) |
| ***–*** | минус (изменение знака, вычитание, разность множеств) |
| **\*** | звездочка (умножение, пересечение множеств) |
| **/** | наклонная черта, слэш (знак деления, результат всегда имеет вещественный тип) |
| **<** | меньше |
| **>** | больше |
| **=** | равно |
| **.** | точка (признак конца программы и модуля) |
| **;** | точка с запятой (разделитель операторов программы) |
| **( )** | левая и правая скобки при выделении подвыражений |
| **<=** | меньше или равно |
| **>=** | больше или равно |
| **<>** | не равно |
| **:=** | операция присваивания |
| **^** | знак карата (обращение к динамической переменной) |
| **And** | операция поразрядного логического сложения (И) |
| **Not** | операция поразрядного дополнения (НЕ) |
| **Or** | операция поразрядного логического сложения (ИЛИ) |
| **Xor** | операция поразрядного логического исключающего ИЛИ |
| **Div** | целочисленное деление |
| **Mod** | остаток от целочисленного деления |
| **Shl** | операция сдвига влево |
| **Shr** | операция сдвига вправо |
| **In** | операция проверки вхождения элемента в множество |
| **Begin…End** | составной оператор |
| **Break** | оператор безусловного выхода из цикла |
| **Continue** | оператор передачи управления на конец тела цикла |
| **Goto <Метка>** | оператор безусловного перехода |
| **Case…Of…Else…End** | оператор варианта |
| **If…Then…Else** | оператор условного перехода |
| **Repeat…Until** | оператор цикла с постусловием |
| **While…Do** | оператор цикла с предусловием |
| **For…To…Do** | оператор цикла с параметром (с увеличением параметра) |
| **For…Downto…Do** | оператор цикла с параметром (с уменьшением параметра) |
| **With…Do** | оператор присоединения |

**Пример 1.** Расчет метрик Холстеда для программы, вычисляющей значение функции

***Y = sin X***

через разложение функции в бесконечный ряд



с точностью ***Eps = 0,0001***.

Текст программы, написанной на языке Паскаль, приведен ниже.

*Program Sin1;*

*Const*

*eps = 0.0001;*

*Var*

*y, x****:*** *real; n****:*** *integer; vs****:*** *real;*

*Begin*

*Readln (x);*

*y := x;* {Начальные установки}

*n := 2;*

*vs := x;*

***Repeat***

*vs := –vs \* x \* x / (2 \* n – 1) / (2 \* n –2);* {Формирование слагаемого}

*n := n + 1;*

*y := y + vs*

***Until abs(vs) < eps;* {Выход из цикла по выполнению условия}**

*Writeln (x, y, eps)*

*End.*

Расчет базовых метрик Холстеда для данной программы приведены в табл. 2.

Таблица 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***j*** | **Оператор** | ***f1j*** | ***i*** | **Операнд** | ***f2i*** |
|  | ; | 7 |  | x | 6 |
|  | := | 6 |  | n | 5 |
|  | \* | 4 |  | vs | 5 |
|  | – | 3 |  | y | 4 |
|  | / | 2 |  | 2 | 4 |
|  | ( ) | 2 |  | 1 | 2 |
|  | + | 2 |  | eps | 2 |
|  | Begin…End | 1 |  |  |  |
|  | Readln ( ) | 1 |  |  |  |
|  | Repeat…Until | 1 |  |  |  |
|  | abs( ) | 1 |  |  |  |
|  | < | 1 |  |  |  |
|  | Writeln ( ) | 1 |  |  |  |
|  | **.** | 1 |  |  |  |
| **1 *=* 14** |  | ***N1 =* 33** | **2 *=* 7** |  | ***N2 =* 28** |

Словарь программы =14 + 7 = 21.

Длина программы ***N*** = 33 + 28 = 61.

Объем программы ***V*** = 