

## Практическое занятие № 14

### «Использование метрик стилистики»

**Цель:** - научиться использовать метрики стилистики и понятности программ.

**Время выполнения:** 2 часа

#### *Теоретические сведения*

Следующие пять характеристик являются продолжением метрики Холстеда.

1. Для измерения теоретической длины программы  $N'$  М.Холстед вводит аппроксимирующую формулу:

$$N' = n_1 * \log_2(n_1) + n_2 * \log_2(n_2)$$

где  $n_1$  - словарь операторов;

$n_2$  - словарь операндов программы.

Вводя эту оценку, Холстед исходит из основных концепций теории информации, по аналогии с которыми частота использования операторов и операндов в программе пропорциональна двоичному логарифму количества их типов. Таким образом, выражение представляет собой идеализированную аппроксимацию  $N$ , т. е. справедливо для потенциально корректных программ, свободных от избыточности или несовершенств (стилистических ошибок). Несοвершенствами можно считать следующие ситуации:

а) последующая операция уничтожает результаты предыдущей без их использования;

б) присутствуют тождественные выражения, решающие совершенно одинаковые задачи;

в) одной и той же переменной назначаются различные имена и т. п.

Подобные ситуации приводят к изменению  $N$  без изменения  $n$ .

М. Холстед утверждает, что для стилистически корректных программ отклонение в оценке теоретической длины  $N'$  от реальной  $N$  не превышает 10%.

Можно использовать  $N'$  как эталонное значение длины программы со словарем  $n$ . Длина корректно составленной программы  $N$ , т. е. программы, свободной от избыточности и имеющей словарь  $n$ , не должна отклоняться от теоретической длины программы  $N'$  более чем на 10%. Таким образом, измеряя  $n_1$ ,  $n_2$ ,  $N_1$  и  $N_2$  и сопоставляя значения  $N$  и  $N'$  для некоторой программы, при более чем 10%-ном отклонении можно говорить о наличии в программе стилистических ошибок, т. е. несовершенств.

На практике  $N$  и  $N'$  часто существенно различаются.

2. Другой характеристикой, принадлежащей к метрикам корректности программ, по М. Холстеду, является уровень качества программирования  $L$  (уровень программы):

$$L = V'/V$$

$V = N * \log_2 n$  — объем программы,

$V' = N' * \log_2 n'$  — теоретический объем программы, где  $n'$  — теоретический словарь программы.

Исходным для введения этой характеристики является предположение о том, что при снижении стилистического качества программирования уменьшается содержательная нагрузка на каждый компонент программы и, как следствие, расширяется объем реализации исходного алгоритма. Учитывая это, можно оценить качество программирования на основании степени расширения текста относительно потенциального объема  $V'$ . Очевидно, для идеальной программы  $L=1$ , а для реальной - всегда  $L < 1$ .

3. Нередко целесообразно определить уровень программы, не прибегая к оценке ее теоретического объема, поскольку список параметров программы часто зависит от реализации и может быть искусственно расширен. Это приводит к увеличению метрической характеристики качества программирования. М. Холстед предлагает аппроксимировать эту оценку выражением, включающим только фактические параметры, т. е. параметры реальной программы:

$L' = (2n_2) / (n_1 * N_2)$  - уровень качества программирования, основанный лишь на параметрах реальной программы без учета теоретических параметров,

4. Располагая характеристикой  $L'$ , Холстед вводит характеристику  $I$ , которую рассматривает как интеллектуальное содержание конкретного алгоритма, инвариантное по отношению к используемым языкам реализации:  $I = L' * V$ .

По мнению самого автора, термин интеллектуальность не совсем удачен. Преобразуя выражение можно получить:

$$I = L' * V = L * V = V' * V / V = V'$$

Эквивалентность  $I$  и  $V'$  свидетельствует о том, что мы имеем дело с характеристикой информативности программы.

Введение характеристики  $I$  позволяет определить умственные затраты на создание программы. Процесс создания программы условно можно представить как ряд операций:

- 1) осмысление предложения известного алгоритма;
- 2) запись предложения алгоритма в терминах используемого языка программирования, т. е. поиск в словаре языка соответствующей инструкции, ее смысловое наполнение и запись.

Используя эту формализацию в методике Холстеда, можно сказать, что написание программы по заранее известному алгоритму есть  $N'$ -кратная

выборка операторов и операндов из словаря программы  $n$ , причем число сравнений (по аналогии с алгоритмами сортировки) составит  $\log_2(n)$ .

Если учесть, что каждая выборка-сравнение содержит, в свою очередь, ряд мысленных элементарных решений, то можно поставить в соответствие содержательной нагрузке каждой конструкции программы сложность и число этих элементарных решений. Количественно это можно характеризовать с помощью характеристики  $L$ , поскольку  $1/L$  имеет смысл рассматривать как средний коэффициент сложности, влияющий на скорость выборки для данной программы. Тогда оценка необходимых интеллектуальных усилий по написанию программы может быть измерена как.

### **Задание:**

1. Написать простое консольное приложение на языке программирования C#, в котором будет реализовано обращение к собственному методу, например вычисляющему несложное арифметическое выражение, создание объекта, и осуществляется вывод текстовых сообщений. (Возможно использование текстов программ, разработанных в рамках других дисциплин, например Калькулятор из ОП.09 Основы алгоритмизации и программирования.)
2. Используя теоретический материал, рассчитать метрику уровня комментированности программы, разбив исходный код на равные части. Составить отчет о проделанной работе.

### **Контрольные вопросы:**

1. Какие метрики можно отнести к метрикам стилистики и понятности программ?
2. Что такое уровень качества программирования?
3. В чем суть метрики корректности программ по М. Холстеду?
4. Привести аппроксимирующую формулу для определения уровня программы.