Федеральное государственное автономное

образовательное учреждение

высшего профессионального образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Кафедра «Информатики» |
| кафедра |

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метод Function Points |
| Тема |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  | И.В. Евдокимов | / |  |
|  | дата |  | инициалы, фамилия |  | подпись |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | КИ14-16Б |  |  |  | С.В. Кухаренко | / |  |
|  | код (номер) группы |  | дата |  | инициалы, фамилия |  | подпись |

Красноярск 2015

**Цель работы**

Научиться производить расчет объема программного проекта и времени разработки с помощью метода функциональных точек.

**Ход работы**

В методике функциональных указателей для определения объема работ используется 5 информационных характеристик:

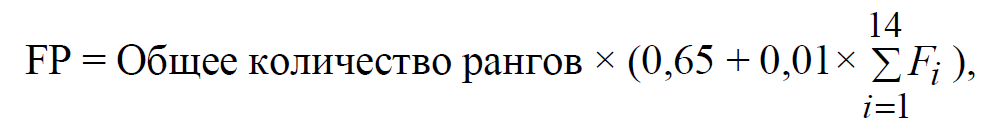
1. Внешний ввод – элементарный процесс, перемещающий данные из внешней среды в приложение. Данные могут поступать с экрана ввода или из другого приложения. Данные могут использоваться для обновления внутренних логических файлов. Данные могут содержать как управляющую, так и деловую информацию. Управляющие данные не модифицируют внутренние логические файлы;
2. Внешний вывод – элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду. Кроме того, в этом процессе могут обновляться внутренние логические файлы. Выводы означают отчеты, экраны, распечатки, сообщения об ошибках или выходные файлы, посылаемые другим приложениям. Отчеты и файлы создаются на основе внутренних логических файлов и внешних интерфейсных файлов. Дополнительно этот процесс может использовать вводимые данные: критерии поиска либо параметры, не поддерживаемые внутренними логическими файлами. Вводимые данные носят временный характер;
3. Внешний запрос – элементарный процесс, работающий как с вводимыми, так и с выводимыми данными. Его результат – данные, возвращаемые из внутренних логических файлов и внешних интерфейсных файлов. Входная часть процесса не модифицирует внутренние логические файлы, а выходная часть не несет данных, вычисляемых приложением (в этом и состоит отличие запроса от вывода);
4. Внутренний логический файл – распознаваемая пользователем группа логически связанных данных, которая размещена внутри приложения и обслуживается через внешние вводы;
5. Внешний интерфейсный файл – распознаваемая пользователем группа логически связанных данных, которая размещена внутри другого приложения и поддерживается им. Внешний файл данного приложения является внутренним логическим файлом в другом приложении.

В ходе данной практической работы проведем расчет объема программного продукта, создаваемого в рамках командного курсового проекта. Среда разработки Microsoft Visual Studio 2015 Community. Язык разработки C#.

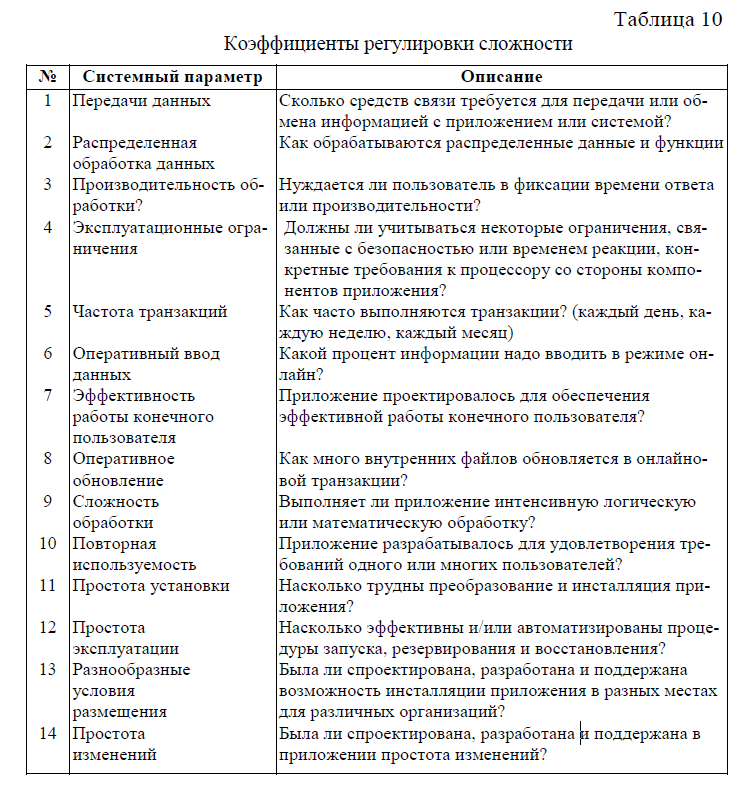
Таблица 1 – Исходные данные для расчета FP-метрик

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя характеристик** | **Ранг, сложность, количество** | | | |
| **Низкий** | **Средний** | **Высокий** | **Итого** |
| Внешние вводы | 11 | 4 | 0 | 148 |
| Внешние выводы | 10 | 8 | 0 | 80 |
| Внешние запросы | 1 | 3 | 0 | 15 |
| Внутренние логические файлы | 0 | 10 | 1 | 115 |
| Внешние интерфейсные файлы | 0 | 0 | 1 | 100 |
| Общее количество рангов | 458 | | | |

Количество функциональных указателей вычисляется по формуле



где Fi – это коэффициенты регулировки сложности (табл. 10), принимающие целые значения: 0 – 5в зависимости от сложности реализации соответствующей характеристики ПП.



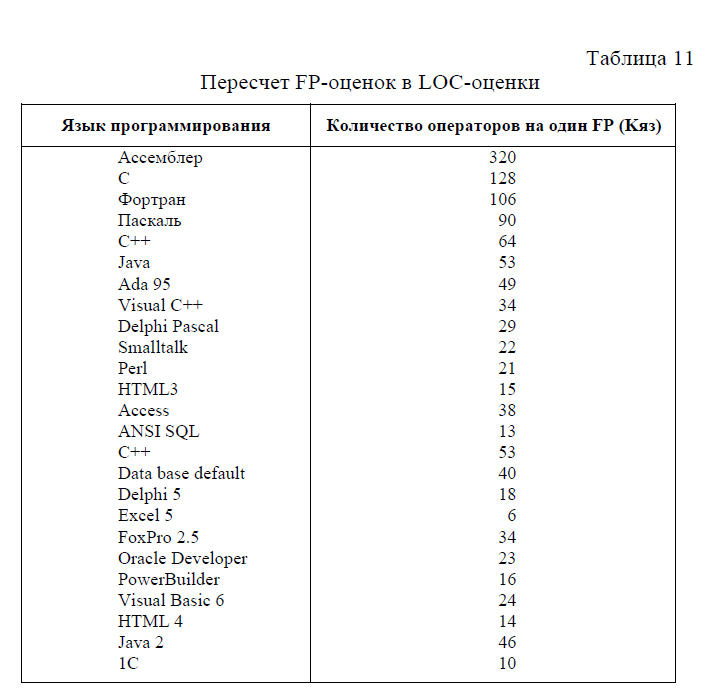
Рассчитаем количество функциональные указателей

FP = 458 × (0,65 + 0,01 × (1+0+0+0+5+3+4+3+1+5+1+1+2+4)) = 435,1

Полученная FP-оценка пересчитывается в LOC-оценки V

V = Kяз × FP,

где Kяз – зависит от языка программирования, используемого для реализации ПО, берется из табл. 11



В данном случае используется язык программирования С++, поэтому расчет будет следующим

V = 53 × 435,1 = 23060,3 LOC

Для перерасчета объема программы в условных строках V в трудозатраты T используются промежуточная модель COCOMO, в соответствии с которой номинальную трудоемкость (без учета коэффициентов затрат труда, стоимостных факторов и сложности) можно вычислить по формуле

Т = N1 × KSLOCN2,

где KSLOC (тыс. строк) = V / 1000, а значения N1 и N2 определяются в зависимости от типа ПО.

T = 3,2 × 23,06031,05­ = 86,33 чел. × мес.

Время разработки вычисляется по формуле

Tразр. = 2,5 × TN3***­*** = 2,5 × 86,330,38­­ = 13,6 мес.

Рекомендуемое правило распределения затрат проекта – 40−20−40:

- на анализ и проектирование приходится 40% затрат (из них на планирование и системный анализ – 5%);

- на кодирование – 20%;

- на тестирование и отладку – 40%.

Таблица 12 – Распределение временных затрат

|  |  |
| --- | --- |
| Этап | Доля затрат, мес. |
| Анализ и проектирование | 5,44 |
| Кодирование | 2,72 |
| Тестирование и отладка | 5,44 |
| Общие временные затраты | 13,6 |

***Список использованных источников***

1. Вендров, А. М. Проектирование программного обеспечения эконо-мических информационных систем: учебник для студ. экон. вузов, обу-чающихся по спец. "Прикл. информатика (по обл.)" и "Прикл. математика и информатика" / Вендров А. М. − 2-e изд., перераб. и доп. − М.: Финансы и статистика, 2006. − 543 с.: ил.

2. Вендров, А. М. Практикум по проектированию программного обес-печения экономических информационных систем: учебное пособие для ву-зов / Вендров А. М. − 2-e изд., перераб. и доп. − М.: Финансы и статистика, 2006. − 191 с.: ил.

3. Орлов, С. А. Технологии разработки программного обеспечения. Современный курс по программной инженерии: учебник для вузов / Орлов С. А., Цилькер Б. Я. − 4-e изд. − Санкт-Петербург [и др.]: Питер, 2012. − 608 с.: ил. (Стандарт третьего поколения).