Расчет по методу функциональных точек

Источник: https://lektsii.org/6-67755.html

Цель работы

Познакомиться с концепциями оценки размера и сложности ПС методом функциональных точек.

Расчет по методу функциональных точек

Рассмотрим пример расчета количества функциональных точек для ПП, реализующего функции телефонного справочника. Допустим, что в справочнике необходимо хранить номер телефона, фамилию и инициалы владельца, а также его адрес. Должны быть предусмотрены возможности поиска записей и сортировки списка. При нажатии на кнопки «Добавить» или «Удалить» выводится сообщение, подтверждающее факт добавления или удаления записи. При попытке добавить запись с существующим номером телефона выводится уведомляющее сообщение.

При выводе списка записи сортируются по номеру телефона, по фамилии или по адресу.

В случае отсутствия искомой информации при поиске выводится соответствующее сообщение.

Внешний вид единственной экранной формы программного приложения приведен на рис. 1.

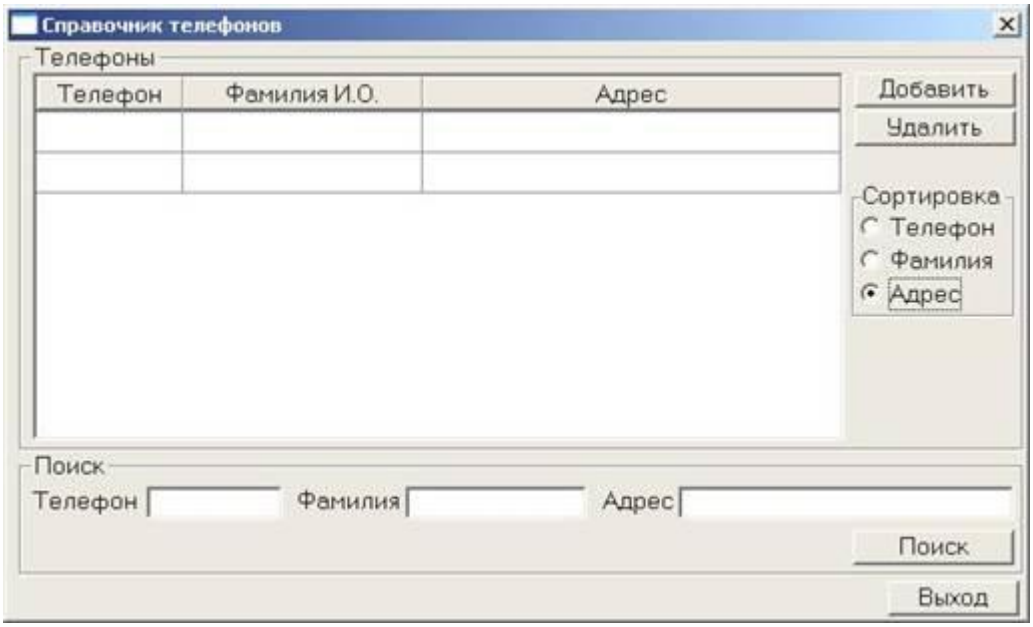

Рис. 1. Экранная форма телефонного справочника

Произведем подсчет количества функциональных точек.

1. ПО является полностью локальным, и обмен данными с другими ПП в нем не предусматривается.

2. В ПО имеется один внутренний логический файл (ILF) для хранения информации справочника. Причем, данные могут храниться как в обычном файле, так и в таблице СУБД.

Число типов элементов записей (RET) для этого файла может быть равно единице или двум. Если данные в файле хранятся в виде однотипных записей: «Телефон», «Фамилия» и «Адрес», которые допустим, представлены в символьном формате, то число RET будет равно 1. В случае же, если номер телефона будет представлен, как целое число, а фамилия и адрес в символьном формате, то тогда внутренний логический файл будет иметь два RET. Для определенности далее будем считать, что внутренний логический файл имеет два RET.

Число типов элементов данных (DET) внутреннего логического файла будет равно трем вне зависимости от формата представления номера телефона («Телефон», «Фамилия», «Адрес»). Таким образом, уровень сложности внутреннего логического файла – низкий.

Внешних интерфейсных файлов (EIF) данное ПО не имеет.

3. В ПП имеются два внешних ввода (EI): «Добавление записи» и «Удаление записи», поскольку именно эти две функции программного продукта модифицируют данные во внутреннем логическом файле. Так как внешний ввод «Добавление записи» ссылается на один внутренний логический файл (т.е. FTR = 1) и имеет пять элементов данных (поля «Телефон», «Фамилия», «Адрес», кнопка «Добавить» и сообщение, подтверждающее факт добавления записи), то уровень сложности этого ввода низкий. Аналогично, уровень сложности внешнего ввода «Удаление записи» также низкий, поскольку имеется один FTR и пять DET (поля «Телефон», «Фамилия», «Адрес», кнопка «Удалить» и сообщение, подтверждающее факт удаления записи).

В программе имеются два внешних запроса (EQ): «Вывод списка» отсортированных записей и «Поиск записи» в справочнике. Внешний запрос «Вывод списка» имеет низкий уровень сложности, так как ссылается на один внутренний логический файл и имеет четыре элемента данных («Телефон», «Фамилия», «Адрес» и группа радио-кнопок «Сортировка»). Уровень сложности «Поиска записи» в справочнике также низкий (один внутренний логический

файл и пять элементов данных: «Телефон», «Фамилия», «Адрес», кнопка «Поиск», сообщение об отсутствии искомой информации).

В ПО имеется также один внешний вывод (EO): вывод уведомляющего сообщения при попытке добавить запись с существующим номером телефона. Уровень сложности этого внешнего вывода – низкий, так как он имеет один FTR и два DET: номер телефона и само сообщение.

Полученные данные сведем в табл. 1. и рассчитаем ненормированное количество функциональных точек.

Характеристика	Количество	Уровень сложности / низкий ранг	Итого
Внешние вводы (EI)	2	3	6
Внешние выводы (EO)	1	4	4
Внешние запросы (EQ)	2	3	6
Внутренние логические файлы (ILF)	1	7	7
Внешние интерфейсные файлы (EIF)	0		0
		Итого:	23

Таблица 1. Данные для расчета ненормированного числа функциональных точек телефонного справочника

4. Подсчитаем теперь итоговую степень влияния (TDI) общих характеристик системы и нормирующий фактор (VAF).

В результате выясним, что для телефонного справочника важны следующие характеристики:

- «Диалоговый ввод данных» (Оперативный ввод данных), который оценивается с весом – 5, поскольку все 100% транзакций в ПС являются интерактивными;

- «Эффективность для конечного пользователя» (Эффективность работы конечных пользователей), которая оценивается с весом – 1, поскольку в ПП имеются функции автоматической установки курсора, скроллинга и интерфейс с мышью;

- «Простота использования» (Легкость эксплуатации), которая оценивается с весом – 5, поскольку в ПС все функции автоматизированы за исключением загрузки/выключения и имеется автоматическое восстановление после ошибок;

- «Распространенность» (Количество возможных установок на различных платформах), которая оценивается с весом – 2, поскольку программное средство рассчитано на работу на совместимом аппаратном/программном обеспечении;

- «Легкость изменения» (Оперативное обновление), которая оценивается с весом – 2, поскольку ПП хранит информацию в таблицах, поддерживаемых пользователем в диалоговом режиме.

Остальные характеристики либо не присутствуют, либо не имеют значения для данного ПО и поэтому имеют вес равный 0.

Нормирующий фактор (VAF) определится как:

$$VAF = 0.65 + 0.1 \cdot TDI = 0.65 + 0.1 \cdot (5+1+5+2+2) = 0.8$$

5. Таким образом, нормированное количество функциональных точек для телефонного справочника вычисляется по формуле:

$$AFPC = UFPC \cdot VAF = 23 \cdot 0.8 = 18.4$$

В заключение, оценим количество строк исходного кода, исходя из того, что программу необходимо разработать с использованием языка программирования C++:

$$SLOC = 18.4 \cdot 53 = 975.2 \approx 975$$

Таким образом, законченная программа телефонного справочника будет содержать примерно 975 строк исходного кода на языке программирования C++.