

Задача №2. Наследование. Использование исключений.

5а (25 баллов)

Требуется реализовать приложение Workflow Executor.

Workflow – вычислительная схема, состоящая из предопределенного набора вычислительных блоков и связей между ними. Программе подается workflow, описанный в файле.

В данной задаче рассматривается единственный тип workflow – линейный, т.е. конвейер.

Список блоков, используемых в схеме:

1. `readfile <filename>` – считывание текстового файла в память, целиком.
Вход – отсутствует, выход – текст.
2. `writefile <filename>` – запись текста в файл.
Вход – текст, выход – отсутствует.
3. `grep <word>` – выбор из входного текста строк, разделенных символами переноса строки, содержащих заданное слово `<word>`.
Вход – текст, выход – текст.
4. `sort` – лексикографическая сортировка входного набора строк.
Вход – текст, выход – текст.
5. `replace <word1> <word2>` – замена слова `word1` словом `word2` во входном тексте.
Вход – текст, выход – текст.
6. `dump <filename>` - сохранить пришедший текст в указанном файле и передать дальше.

Формат входного файла:

```
desc # описание блоков схемы
id1 = block1
id2 = block2
...
idN = blockN
csed
idA -> idB -> idC -> ... -> idZ # описание структуры схемы
```

Где:

- `desc`, `csed` – ключевые слова, ограничивающие раздел описания блоков workflow.
- `id1 ... idN` – целые, неотрицательные, неповторяющиеся числа.
- `block1 ... blockN` – блоки из списка блоков, с обязательными параметрами.
- `idA, idB ... idZ` – числа, принадлежащие множеству `id1...idN`. Могут повторяться, длина конвейера – неограничена.
- `->` – ключевое слово, обозначающее связь вычислительных узлов.

Ограничения на вычислительную схему:

1. `id1...idN` – не повторяются.
2. Блоки должны содержать нужное количество параметров.

3. В описании структуры схемы первый блок должен быть блоком чтения из файла, последний – блоком записи в файл.
4. Узлы схемы должны корректно соединяться, то есть тип данных на входе узла должен совпадать с типом данных предыдущего узла. Это означает, что чтение/запись файлов не может быть в середине схемы.

Пример:

```
workflow.txt
desc
1 = replace abracadabra cadabraabra
2 = grep braab
3 = sort
0 = readfile in.txt
5 = writefile out.txt
csed
0 -> 1 -> 2 -> 3 -> 5
```

Запуск:

```
executor.exe workflow.txt
```

Методические указания по реализации задачи:

1. Создать интерфейс Worker. Классы представляющие блоки схемы должны его имплементировать.
2. Для парсера схем и вычислителя/валидатора также выделить отдельные интерфейсы.
3. Для обработки ошибок и исключительных ситуаций использовать механизм исключений C++. Ознакомиться с классами исключений, определенных в STL.
4. Для работы с файлами использовать файловые потоки из STL.
5. При работе с потоками ввода-вывода, за состоянием потока наблюдать с помощью исключений. См. метод `std::ios::exceptions` (<http://cplusplus.com/reference/iostream/ios/exceptions>).

При сдаче задания продемонстрировать пять рабочих и пять нерабочих workflow.

5б (5 баллов)

Добавить поддержку конвейеров, возможно не содержащих блока чтения файла и/или блока записи файла. В случае, если какой-либо из них не указан, ожидать соответствующих имен из командной строки под ключами `-i` и `-o`.

Пример:

```
workflow.txt:
desc
1 = grep some_word
2 = sort
csed
1 -> 2
```

Запуск:

```
executor.exe workflow.txt -i input.txt -o output.txt
```