Упражнение 1 (Разбор параметров командной строки )

Задача состоит в написании класса, который разбирает командную строку, переданную на вход программе. При создании объекта этого класса, он получает набор разделителей и набор самих ключей, который допустим в командной строке. Например: «-t -r», здесь «-» это разделитель, а «t» и «r» ключи. Класс содержит два вспомогательных метода:

* OnUsage - выводит подсказку с форматом командной строки;
* OnSwitch – отрабатывает команду для каждого найденного ключа;

Также в классе пишем метод, содержащий бизнес логику по разбору командной строки – метод “Parse”.

- Создайте новый проект в среде «NetBeans».

- Создайте новый public класс с именем «ComLineParser».

- Объявите массив строк для хранения ключей, которые могут встретиться в командной строке - «keys».

- Объявите второй массив строк для хранения разделителей, с которых и будут начинаться ключи в командной строке - «delimeters».

- Объявите следующее перечисление в классе:

// варианты завершения разбора командной строки

public enum SwitchStatus { NoError, Error, ShowUsage };

- Создайте public конструктор с двумя параметрами. Первый, массив строк ключей. Второй, массив строк разделителей. Инициализируйте

члены класс этими значениями.

- Создайте второй public конструктор, но с одним параметром – массив ключей. В теле конструктора переключите выполнение кода на

конструктор с двумя параметрами, где вторым параметр будет массив из двух строк «/» и «-».

- Создайте метод для вывода информации об ошибочном ключе и о вариантах командной строки:

* объявите public метод с именем «OnUsage» без возвращаемого значения и с одним строковым параметром «errorKey»
* если «errorKey» не равно null, то печатаем его
* печатаем информацию о работе командной строки

System.out.println("формат ком.строки: имяПрограммы [-r<input-fileName>] [-w<output-fileName>]");

System.out.println(" -? показать Help файл");

System.out.println(" -r задать имя входного файла");

System.out.println(" -w выполнить вывод в указанный файл");

- Создайте метод, который будет вызываться для каждого найденного ключа в командной строке:

* объявите public метод с именем «OnSwitch», возвращает «SwitchStatus» и получает на вход две строки «key» и «keyValue»

(имя найденного ключа в командной строке и его значение соответственно)

* в теле метода распечатайте оба параметра и верните «SwitchStatus .NoError»

- Создайте метод, который и будет выполнять разбор параметров командной строки:

* объявите public метод с именем «Parse», который возвращает логический тип и принимает на вход массив параметров командной строки – «args»
* в методе устанавливаем статус в значение «NoError» и запускаем цикл по элементам массива «args»

SwitchStatus ss = SwitchStatus.NoError;

int argNum;

for (argNum = 0; (ss == SwitchStatus.NoError) && (argNum < args.length); argNum++) {}

* в теле цикла для каждого параметра командной строки проверяем, что он начинается с одного из допустимых разделителей

// провера наличия правильного разделителя

boolean isDelimeter = false;

for (int n = 0; !isDelimeter && (n < delimeters.length); n++) {

isDelimeter = args[argNum].regionMatches(0,delimeters[n], 0, 1);

}

* ниже напишите условный оператор, который при правильном разделителе начнёт проверку на наличие допустимого ключа, а в противном случае

завершит алгоритм со статусом «Error»

if (isDelimeter) {

// проверка наличия правильного ключа

......

}

else {

ss= SwitchStatus.Error;

break;

}

* после комментария // проверка наличия правильного ключа

объявите логическую переменную «isKey» с начальным значением false и целочисленную переменную «i»

* начните цикл по ключам

for (i = 0; !isKey && (i < keys.length); i++) {}

* в теле цикла присвойте переменной «isKey» результат сравнения двух ключей предварительно приведя их к верхнему регистру(используйте метод

regionMatches). Первый ключ для сравнения берётся из массива «args», причём не с нулевой позиции, а второй и массива «keys»

* если окажется, что ключ верен, то завершаем цикл оператором «break»
* после закрывающей скобки цикла по «i» проверяем значение переменной «isKey» и если окажется, что оно «false», то завершаем алгоритм со статусом «Error»
* после закрывающей скобки цикла по «argNum», в случае наличия ошибок, выводим информацию о параметрах командной строки

// завершение разбора командной строки

if (ss == SwitchStatus.ShowUsage) OnUsage(null);

if (ss == SwitchStatus.Error) OnUsage((argNum == args.length) ? null : args[argNum]);

* завершаем алгоритм написав

return ss == SwitchStatus.NoError;

- В свойствах проекта, в категории «Выполнить», в графе «Аргументы», задайте командную строку: -wOutfileName -rInfileName

-В методе «main» создайте объект класса «ComLineParser» со следующим набором ключей "?","r","w" и разделителями по умолчанию.

- Вызовите на этом объекте метод «Parse», передав ему массив параметров «args».

- Запустите программу.