

Тема: наследование и полиморфизм подтипов

Вариант: 2.2.1

Задача: Реализовать универсальную сортировку натуральных чисел.

Реализовать абстрактный класс порядков `Order`, являющийся корнем иерархии.

Класс `Order` должен содержать чисто виртуальный метод отвечающий на вопрос “Правильно ли стоят два данных числа в данном порядке?”

На базе класса `Order` реализовать классы-наследники:

1. `Ascending` и `Descending` соответствующие порядкам по возрастанию и по убыванию, соответственно.
2. Класс `ByLength` соответствующий упорядочиванию по длине десятичной записи числа. Конструктор класса `ByLength` должен принимать параметром другой порядок, используемый для упорядочения длин.
3. Класс `ByBitCount` представляющий порядок по числу ненулевых бит в двоичной записи числа. При конструировании принимает параметром дополнительный порядок, используемый для упорядочения числа бит.

Пример:

```
Order * o = new ByLength(new Descending());
```

Здесь `o` соответствует упорядочению по убыванию длины десятичной записи числа.

В качестве демонстрационного примера необходимо:

1. Считать из входного файла описание порядка и массив чисел.
2. Записать в выходной файл массив отсортированный в указанном порядке. Сортировка должна быть устойчивой, то есть не должна менять порядок равных (несравнимых) элементов.

Входные данные:

В первой строке входного файла закодирован порядок:

- Символ A -- порядок по возрастанию
- Символ D -- порядок по убыванию
- Строка Lxxx -- порядок “по длине”, xxx кодирует дополнительный порядок на длинах.
- Строка Vxxx -- порядок по числу бит, xxx кодирует дополнительный порядок.

Во второй строке записано число N --- длина массива данных.

В третьей строке через пробел перечислены N натуральных чисел.

Выходные данные:

В выходной файл записать данные числа отсортированные в заданном порядке.

Пример входных и выходных данных:

input.txt	output.txt
A 6 16 4 15 42 23 8	4 8 15 16 23 42
BD 10 1 2 3 4 5 6 7 8 9 10	7 3 5 6 9 10 1 2 4 8