

HW2013PY

Внутри ru-файлов оставлять комментарии - имя фамилия, номер группы, номер задания.

- 01 Вывести таблицу сложения для заданной цифры k (любой цикл).
Т.е. на экране должно быть (например, для 3):

```
Enter digit: 3
3x2 = 6
3x3 = 9
3x4 = 12
3x5 = 15
...
3x9 = 27
```

- 02 Подсчитать для введенного x сумму ряда с точностью $eps = 0.000000001$

$$\sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{(2i)!}$$

- 03 Вводится целое число n . Вывести ромб с горизонтальной диагональю равной $2n + 1$ по следующему образцу (например, для $n=5$)

```
*****0*****
****000****
***00000***
**0000000**
*000000000*
00000000000
*000000000*
**0000000**
***00000***
****000****
*****0*****
```

- 04 На вход программы подается файл in.txt. Получать файл out.txt, в котором все буквы английского алфавита заменены на свои заглавные аналоги, а заглавные - на строчные.
- 05 Написать регулярные выражения для

- телефона в формате:

8 (123) 456-78-90

- телефона в формате:

+7-123-456-78-90

Используя их, написать программу, которая проверяет, что во введенном массиве строк все строки представляют собой корректные телефонные номера (т.е. удовлетворяют либо первому, либо второму регулярному выражению).

- 06 Реализовать класс RationalFraction - Рациональная дробь, т.е. дробь вида m/n , где m, n - целые, содержащий следующие методы:

Конструкторы с параметрами и без.

toString()

все необходимые set и get

возврат значения дроби в виде десятичного числа

сложение дробей (с возвращением результата) (оператором)

умножение дробей (с возвращением результата) (оператором)

- 07 Реализовать класс Vector2D - двумерный вектор, содержащий следующие методы:

Конструкторы с параметрами и без.

toString()

все необходимые set и get

возврат длины вектора

скалярное произведение с другим вектором

сумма с другим вектором (вернуть результат в виде нового вектора) (оператором)

- 08 Реализовать класс VectorRF2D - двумерный вектор, элементами которого являются экземпляры класса «Рациональная дробь» (55). Вектор должен содержать те же методы, что и вектор из задачи 56, но в результатах везде использовать не числа, а RationalFraction. Если потребуется реализовать в RF дополнительные методы (умножение, например), сделайте это.

- 09 Дан файл input.txt, в котором записаны целые числа.

Например: 1 200 -199 2 234 2 1 34 7 ... и т.д.

Вывести на экран данные о количестве вхождений каждого числа в исходный файл, используя словарь. Т.е., на экране должно быть:

1 : 2 раз

200 : 1 раз

-199 : 1 раз

2 : 2 раз

и т.д.

- 10 Во входном файле представлен текст на английском языке. Он содержит слова, записанные через пробел, не содержащие в себе иных символов, кроме алфавита. Но слова могут быть записаны разным регистром.

Создать словарь, содержащий информацию о том, сколько раз каждое уникальное слово из файла встречается в нем. Словарь не должен зависеть от регистра слова, т.е. слова One, OnE, ONE и one - одно и то же слово.

Содержимое полученного словаря вывести на экран.

10.5 СОЗДАТЬ 30 ФАЙЛОВ

ИМЯ ФАЙЛА: СЛУЧАЙНОЕ ЧИСЛО ОТ 10 ДО 50

РАСШИРЕНИЕ ФАЙЛА: СЛУЧАЙНАЯ СТРОКА: ЛИБО txt, ЛИБО dat, ЛИБО src

СГЕНЕРИРОВАТЬ ЕЩЕ 30 ИМЕН ФАЙЛОВ С РАСШИРЕНИЯМИ И ПОПЫТАТЬСЯ ИХ ОТКРЫТЬ
ЕСЛИ ТАКОГО ФАЙЛА НЕТ, ТО СОЗДАТЬ ЕГО И ЗАНЕСТИ ТУДА СЛОВО NEW
ЕСЛИ ТАКОЙ ФАЙЛ ЕСТЬ, ТО ОТКРЫТЬ ЕГО, ЗАНЕСТИ ТУДА СЛОВО OLD

ПРОЙТИ ПО ВСЕМ ФАЙЛАМ В ПАПКЕ, С ПОМОЩЬЮ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ ПЕРЕИМЕНОВАТЬ
ФАЙЛЫ С РАСШИРЕНИЕМ txt И НАЧИНАЮЩИЕСЯ С 1 В ФАЙЛЫ С РАСШИРЕНИЕМ src

ПРОЙТИ ПО ВСЕМ ФАЙЛАМ В ПАПКЕ И СОЗДАТЬ СЛОВАРЬ:

"OLD" : [ИМЕНА ФАЙЛОВ, СОДЕРЖАЩИХ OLD]

"NEW" : [ИМЕНА ФАЙЛОВ, СОДЕРЖАЩИХ NEW]

" " : [ИМЕНА ФАЙЛОВ, НЕ СОДЕРЖАЩИХ НИЧЕГО]

- 11 Написать SQL-код создания следующих таблиц с указанием в них первичных (primary) и внешних (foreign) ключей (названия таблиц придумать самостоятельно):

- Таблица «Преподаватели» с полями: код преподавателя (pKod - уникальный ключ), фамилия (pFIO), кафедра (pKaf).
- Таблица «Студенты» с полями: код студента (sKod - уникальный ключ), фамилия (sFIO), группа (sGr).
- Таблица «Предметы» с полями: код предмета (uKod - уникальный ключ), наименование (uNam), количество часов (uKol).
- Таблица «Экзамены»: код студента (sKod - кто сдавал), код предмета (uKod - что), код преподавателя (pKod - кому), дата (eDt - когда), балл (zBall).

Реализовать следующие запросы:

- Выбрать всех студентов, которые сдали хотя бы один экзамен на количество баллов больше 80.
- Отобразить таблицу "Экзамену" в информативном виде, т.е. с указанием фамилии студента, фамилии преподавателя и наименования предмета вместо соответствующих ключей.

Все запросы (включая создание таблиц) сделать из ру-скрипта.

- 12 Файловый менеджер с работающими ls, dir, cd и cat
- 13 Задача с компилированием и выполнением .java файлов.