# Теоретическая часть

* Последовательные бинарные потоки ввода и вывода
* Бинарные потоки ввода/вывода прямого доступа
* Бинарная сериализация и десериализация
* Интерфейс **Serializable**
* Ключевое слово **transient**, запрет сериализации для атрибута класса

# Практическая часть

Напишите консольное Java приложение с простым меню для решения следующих задач.

**Задача 1.** Для коллекции жителей некоторых городов (фамилия, имя, возраст, название города, профессия, оклад) cоставьте и выполните запросы к коллекции с использованием **StreamAPI**. *Храните коллекцию в бинарном файле с фиксированным размером записи. Доступ к файлу – последовательным потоком ввода, последовательным потоком вывода*:

1. статистика по городам – название городов, количество жителей из этих городов в коллекции, средний возраст, минимальный возраст, максимальный возраст жителя
2. жители с заданной профессией, фамилия которых начинается с заданной строки
3. список фамилий и жители с такой фамилией
4. список профессий и жители с такой профессией
5. список городов по убыванию количества проживающих в них людей
6. статистика по профессиям – количество жителей с заданной профессией, минимальный оклад, средний оклад, максимальный оклад, сумма окладов
7. среднее количество жителей города и города с количеством жителей ниже среднего

**Задача 2.**Дан текстовый файл из нескольких строк, слова в котором разделяются пробелами (одним или несколькими), символами конца строки. Реализовать обработки:

* Вывести текстовый файл в консоль, сохраняя разбивку текста на строки
* Преобразовать файл, оставив между словами по одному пробелу
* Для всех слов, из которых состоит текст определить: количество слов, суммарное количество букв в словах, минимальная длина слова, максимальная длина слова, средняя длина слова. *Сохранить статистику в бинарном файле при помощи бинарных потоков прямого доступа. Вывести эту статистику в консоль из файла*

**Задача 3.** Моделирование работы склада. Данные склада хранятся в файле, *созданном бинарной сериализацией (интерфейс* ***Serializable***). По приведенному набору данных разработайте класс **Goods**, описывающий товар и класс **Store** для хранения и обработки коллекции товаров. В таблице столбец Стоимость и строка Итого – вычисляемые, хранить их не надо.

Перечень товаров на складе «Оптовый», г. Донецк, ул. Циолковского, 21

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование товара** | **Количество** | **Цена ед. товара** | **Стоимость** |
| Зонтик автоматический | 50 | 750.00 | 37500.00 |
| Портсигар импортный | 300 | 1200.00 | 360000.00 |
| Приставка игровая | 2 | 15600.00 | 31200.00 |
| Куртка замшевая | 12 | 5100.00 | 61200.00 |
| Чайник фарфоровый | 15 | 750.00 | 11250.00 |
| Кинокамера отечественная | 7 | 3100.00 | 21700.00 |
| Телевизор цветной | 7 | 25100.00 | 175700.00 |
| Фотоаппарат цифровой | 2 | 12000.00 | 24000.00 |
| Компьютер планшетный | 3 | 15600.00 | 46800.00 |
| Кофейник фарфоровый | 15 | 750.00 | 11250.00 |
| **Итого** | **413** |  | **780600.00** |

Требуется выполнение следующих обработок при помощи **StreamAPI**:

* вывод коллекции в консоль с итогами по количеству и сумме товара (сумма товара вычисляется как по строке, так и по столбцу);
* упорядочивание коллекции по убыванию стоимости товара;
* упорядочивание коллекции по наименованию товара;
* упорядочивание коллекции по возрастанию количества товара;
* выполните количественный анализ: суммарное количество товара, минимальное, максимальное, среднее количество товара

# Дополнительно

Запись занятия можно скачать по [**этой ссылке**](https://cloud.mail.ru/public/jFsf/gVVY58oWo). Материалы занятия в этом же архиве.