Данное домашнее задание является общим для 14 и 15 уроков. Задачи 1-15 необходимо решать в области, предназначенной для тестов. Для каждой задачи создать отдельный класс TaskN, в котором будут находиться методы-тесты, в которых и будет решение задачи (каждый пункт задачи – отдельный тестовый метод).

**Задача 0**

В приложенном файле dataset.json содержится информация о клиентах банка. Необходимо создать 2 пакета: model и helpers.

В пакете model реализовать по имеющемуся dataset.json модельные классы. Главный класс назвать Person. В каждом классе реализовать конструктор без параметров, конструктор со всеми параметрами, геттеры и сеттеры для каждого поля.

В пакете helpers создать класс JsonHelper. В данном классе создать статический метод getPersonFromFile(String path), возвращающий тип List<Person>. Данный метод в качестве параметров должен принимать путь к файлу, считывать из файла информацию, содержащуюся о клиентах, десериализовывать в List<Person> и возвращать в качестве результата.

В дальнейших задачах мы будем использовать наш класс-помощник для получения данных. Для каждой следующей задачи получать данные с помощью метода getPersonFromFile, передавая путь к файлу dataset.json, а задачи решать с помощью методов Stream API.

**Задача 1**

Формат вывода: <Фамилия> <Имя> <Отчество>

1. Вывести на экран информацию о тех людях, у которых нет паспорта, но при этом на них зарегистрировано 3 единицы недвижимости. Убедиться, что только 3 клиента подходят под данное условие.

|  |
| --- |
| Ожидаемый вывод |
| Казанцев Василий Иванович  Иванов Сергей Алексеевич  Иванов Григорий Владимирович |

2. Вывести на экран информацию о тех людях, у которых количество счетов в банке не равно количеству карт, фамилия этих людей должна быть «Васильев». Убедиться, что таких клиентов 10.

|  |
| --- |
| Ожидаемый вывод |
| Васильев Алексей Сергеевич  Васильев Григорий Алексеевич  Васильев Сергей Алексеевич  Васильев Алексей Иванович  Васильев Григорий Александрович  Васильев Григорий Григорьевич  Васильев Олег Алексеевич  Васильев Михаил Александрович  Васильев Георгий Алексеевич  Васильев Михаил Владимирович |

3. Найти тех людей, у которых серия паспорта начинается с «00». Вывести на экран только первые 5 записей.

|  |
| --- |
| Ожидаемый вывод |
| Петров Григорий Владимирович  Горбунов Василий Васильевич  Иванов Григорий Георгиевич  Богданов Алексей Георгиевич  Петров Олег Георгиевич |

**Задача 2**

Формат вывода: <ФИО> <Серия паспорта> <Номер паспорта>

1. Найти информацию о всех людях, серии паспортов которых начинаются на «00». Вывести на экран в порядке убывания номера паспорта информацию

|  |
| --- |
| Ожидаемый вывод |
| Кузнецов Иван Васильевич 0005 981574  Петров Григорий Владимирович 0037 930656  Петров Олег Георгиевич 0000 869942  Горбунов Василий Васильевич 0046 655221  Иванов Григорий Георгиевич 0085 388506  Богданов Алексей Георгиевич 0026 029391 |

1. Найти информацию о первом человеке, у которого последние 4 цифры номера паспорта совпадают с номером серии. Вывести на экран информацию о нем. Если такого человека нет, выбросить исключение NoSuchElementException.

|  |
| --- |
| Ожидаемый вывод |
| Иванов Алексей Алексеевич 6483 086483 |

**Задача 3**

1. Необходимо среди всех клиентов посчитать количество различных вариаций серий паспортов. Вывести значение на экран

|  |
| --- |
| Ожидаемый вывод |
| Количество различных серий паспортов: 473 |

1. Среди списка клиентов, у которых есть паспорта, выбрать начиная с 50-го, заканчивая 99-ым включительно. Из выбранных клиентов выбрать первого, имеющего три открытых счета и две карты. Если клиента по такому условию не найдено, взять первого из полного списка. Вывести на экран информацию о клиенте в формате:

<ФИО>

<№ счета 1> <№ счета 2> <№ счета 3>

<№ карты 1> <№ карты 2>

|  |
| --- |
| Ожидаемый вывод |
| Кузьмин Иван Петрович  03993.328.3.16306561347 03443.364.6.62434564962 81821.643.9.65301523431  4593279110041199 5370923335643534 |

1. Проверить, что для каждого клиента, у которого есть паспорт, серия паспорта состоит из 4 цифр, а номер паспорта – из 6 цифр. Проверять необходимо именно цифры, букв в серии и номере быть не может. Вывести результат проверки на экран в одном из форматов:

Если да -> «У всех клиентов, у которых есть паспорт, данные соответствуют формату»

Если нет -> «Есть клиенты, у которых данные паспорта не соответствуют формату»

|  |
| --- |
| Ожидаемый вывод |
| У всех клиентов, у которых есть паспорт, данные соответствуют формату |

**Задача 4**

1. Из полного списка клиентов банка выбрать тех клиентов, у которых есть паспорт, имеется 3 открытых счета и 0 карт. Результат фильтрации сохранить в созданную локальную переменную List<Person> filteredPersons. Убедиться, что размер данного списка равен 0. Для полученного пустого списка проверить следующие утверждения и вывести на экран результат проверки (формат – true/false)

* Для всех элементов отфильтрованного списка серии паспортов соответствуют формату «4 цифры»
* Есть хотя бы один элемент отфильтрованного списка, у которого серия паспорта соответствует формату «4 цифры»
* Нет ни одного элемента отфильтрованного списка, серии паспортов которого соответствуют формату «4 цифры»

|  |
| --- |
| Ожидаемый вывод |
| true  false  true |

1. Создать приватный метод, определяющий, является ли карта действующей на сегодняшний день. Метод должен возвращать логическую переменную: true, если карта действующая, false, если карта истекла. Например, карта со сроком 12/19 считается истекшей 1 декабря 2019 года. В качестве аргумента метод должен принимать объект типа Card. В случае возникновения исключения ParseException метод должен обрабатывать его, выкидывая RuntimeException. Посчитать количество людей, у которых:

* Есть карты
* Есть карты и все они являются действующими
* Есть карты и все они с истекшим сроком действия
* Есть карты и среди них есть и действующие, и истекшие.

Вывести каждый из счетчиков на экран. Убедиться, что счетчик «Есть карты» является суммой трех остальных счетчиков

|  |
| --- |
| Ожидаемый вывод |
| 462  153  152  157 |

**Задача 5**