# Лабораторная работа №2: Немного об ОКТМО

Тема: чтение текстовых данных, регулярные выражения, стандартные коллекции.

Работа заключается в чтении массива данных на примере классификатора ОКТМО (список муниципальных образований и населённых пунктов России) и анализе его содержимого без использования баз данных и специализированных библиотек.

Дан текстовый файл формата .csv с содержимым государственного классификатора ОКТМО

Простейшая работа с системой контроля версий Git:

* Зарегистрируйтесь на сайте gitlab.ugatu.su и дождитесь подтверждения аккаунта администратором (письма редко доходят, похожи на спам)
* Зайдите на страницу <https://gitlab.ugatu.su/java/java-lab2> и создайте свою личную копию этого проекта (форк):



* Найдите проект в разделе Projects/Your projects и скопируйте адрес хранилища.
* Скачайте заготовку проекта с общего хранилища. Варианты:
  + из командной строки (с установленным Git for Windows или в Linux)  
    git clone https://gitlab.ugatu.su/df /java-lab2
  + IDEA: Меню File – New – Project from version control – Git – вставить URL
  + NetBeans: Меню Группа-Git-Клонировать– вставить URL
* После изменений (работающей части лабораторной) можно сохранять текущее состояние проекта – осуществлять фиксацию (коммит)
  + Командная строка – git status , git add файл , git commit -m Сообщение (в папке проекта)
  + Графический инструмент git gui (в папке проекта из командной строки)
  + NetBeans – правой кнопкой на проекте в дереве проектов – Git – Фиксация –проверить измененный файлы, ввести соощение и нажать Фиксировать
  + IDEA – Меню VCS – Commit (или Ctrl-K)
* Один или несколько коммитов можно опубликовать на сервере
  + командой git push origin master
  + IDEA: VCS – Git – Push
  + NetBeans: Группа – Удалённый (remote) – Вытолкнуть (Push)

1. Просмотрите файл .csv в среде разработки или текстовом редакторе.
   1. <https://notepad-plus-plus.org/download/>
   2. Найдите регионы, муниципальные районы, поселения (сельсоветы и др.), городские округа и входящие в них населенные пункты. Обратите внимание на коды ОКТМО (часть цифр отвечает за регион, часть за район/городской округ, дальше – поселения и НП.
   3. Короткие коды у муниципальных образований (их пока нужно игнорировать), длинные – у населённых пунктов.
2. Откройте проект. Добавьте заготовки классов OktmoReader, OktmoData, OktmoAnalyzer, если их нет.
   1. OktmoData будет хранить всю считанную информацию (списки) и содержать методы обращения к ним
   2. OktmoReader считывает текстовые файлы .txt и добавляет их содержимое в разобранном виде в OktmoData
   3. OktmoAnalyzer просматривает объекты в OktmoData и проводит анализ.
3. Чтение списка НП (без учёта иерархии)
   1. Добавьте класс Place (населенный пункт) с полями code (long) и name, status (пока текст)  
      (code= 80 602 418 101 name=Зелёный Клин status=деревня)
   2. Добавьте список объектов Place к классу OktmoData (изначально пустой)
   3. Добавьте метод public void readPlaces(String fileName, OktmoData data) к классу OktmoReader
   4. sample1.txt – пример построчного чтения файла
   5. Для начала примените метод .split к каждой считанной строке (передаём символ-разделитель, получаем массив строк), напечатайте содержимое. Разделитель в csv – точка с запятой ;.
   6. Сконвертируйте первые элементы в числа ( Long.parseLong )  
      (кавычки можно убрать методом replace("\"","") в классе String ) или s.substring(1, s.length()-1). Код собирайте из первых нескольких столбцов
   7. Вызывайте для каждой считанной строки конструктор Place и заносите созданных объект в OktmoData (метод addPlace)
   8. Напечатайте содержимое OktmoData после чтения.  
      (\*) (\* здесь и далее = "по желанию")  
      Другие примеры разбора строки:  
      <http://stackoverflow.com/questions/5965767/performance-of-stringtokenizer-class-vs-split-method-in-java>   
      Попробуйте использовать только indexOf и замерьте время чтения файлa двумя способами.
4. Исправьте ”недоработки” при чтении: отделите статус «д» в отдельное поле (рекомендую методы строки indexOf и substring или поиск первой заглавной буквы по коду символа), пропускайте строки, начинающиеся с “Населенные пункты ….” (метод startsWith у строки).
5. Напишите тест JUnit для чтения (проверяйте кол-во НП и название/статус/код 10-го и последнего в списке НП)
6. Составьте множество статусов (д г пгт и т. д.) – HashSet<String> или TreeSet<String> - поле allStatuses в классе OktmoData . Добавляйте в него элементы при чтении.
   1. Напечатайте после чтения всего объёма данных. В чём отличие HashSet и TreeSet?
7. Отсортируйте список населённых пунктов по имени (метод OktmoData)
   1. Cоздайте отдельное поле sortedPlaces, cкопируйте туда весь ArrayList.
   2. Вызовите стандартную процедуру сортировки Collections.sort. Так как объекты Place изначально не сравнимы друг с другом, придётся написать класс-компаратор  
      <http://echuprina.blogspot.ru/2012/02/comparable-comparator.html>  
      <http://www.leveluplunch.com/java/tutorials/007-sort-arraylist-stream-of-objects-in-java8/>
8. С помощью регулярных выражений найдите все НП, название которых содержит меньше 6 букв и заканчиваются на -ово (метод OktmoAnalyzer)  
   <http://www.quizful.net/post/Java-RegExp>  
   <http://javagu.ru/portal/dt?last=false&provider=javaguru&ArticleId=GURU_ARTICLE_64538&SecID=GURU_SECTION_63111>
9. Найдите населённые пункты, с названиями, которые начинаются и заканчиваются га одну и ту же согласную букву ( использовать [^…] , () , \\1, а также [Pattern.CASE\_INSENSITIVE](http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html#CASE_INSENSITIVE) или (?i) в регулярном выражении +[Pattern.UNICODE\_CASE](http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html" \l "CASE_INSENSITIVE) для русских букв) (метод OktmoAnalyzer)
10. (\*) Создайте альтернативную процедуру чтения списка НП без split и indexOf с использованием ровно одного регулярного выражения с группами (метод OktmoReader)  
    <http://initialize.ru/regular-expressions-in-java>
11. (\*) Создайте тест на чтение файла вторым способом (сравните время работы) и тест на проверку одинаковости данных (используйте equals для коллекций, написав свой equals для класса Place)