**Вариант 7.**

**Показатели спортивных команд.**

Так как мне достался 7 вариант, то я имею следующие задачи:

1. Постройте график по среднему возрасту во всех командах.
2. Найдите команду с самым высоким средним ростом. Выведите в консоль 5 самых высоких игроков команды.
3. Найдите команду, с средним ростом равным от 74 до 78 inches и средним весом от 190 до 210 lbs, с самым высоким средним возрастом.

Я недолго думал: «Как же расписать ход выполнения работы?». И пришел к следующему выводу: расписать для чего нужен каждый класс и какую задачу выполняют содержащиеся в них методы. Таким образом, получилось следующее описание:

1. Для начала я создал класс Sportsman. Он предназначен для записи данных отдельной строки из файла SportTeams.csv.
   1. В классе имеются следующие аргументы: name, team, position, height, weight и последний, но не менее важный – age. Эти аргументы соответствуют столбцам в таблице SportTeams.csv.
   2. В классе также присутствует конструктор и геттеры\сеттеры для имеющихся аргументов.
2. Далее я организовал класс CsvParser для сбора информации из файла SportTeams.csv. Этот класс имеет единственный метод – parse, принимающий путь к файлу и возвращающий список спортсменов (класс Sportsman).
   1. Первым делом я создаю пустой список sportsmans.
   2. Затем я создаю csvReader для чтения информации из файла и создаю массив строк nextLine (необходим для сбора информации из строки SportTeams.csv).
   3. После с помощью цикла while обхожу каждую строку в файле и записываю информацию в sportsman, который добавляю в список sportsmans. Происхлдит это до тех пор, пока строки в файле не закончатся (т.е nextLine == null).
3. Следующий шаг – создание базы данных. Для этого я написал класс DataBase. По сути, я этот класс прописал ранее в работе, которую мы реализовывали в течении семестра, так что я просто скопировал код и немного подправил:
   1. Создаю аргументы connection (для подключения к БД) и ***dbUrl*** (ссылка на БД).
   2. Далее я прописываю конструктор класса, где задаю подключение к БД.
   3. Затем следует метод createTable, из названия понятно, что он служит для создания таблицы в БД. В методе прописан SQL запрос, который при последующем выполнении создаст таблицу Sportsmans с заданными полями внутри БД.
   4. Теперь переходим к методу insertSportsman. Метод необходим для добавления данных в БД. На вход подаётся список классов Sportsman. Далее я прописываю SQL запрос для добавления значений из Sportsman в БД. Следующий шаг – обход всех элементов sportsmans и выполнение запроса.
4. Класс Task – это класс для решения поставленных задач. Задачи указаны выше и каждый метод выполняет соответствующую задачу.
   1. displayChart: просто метод для вывода графика на экран. Не вижу смысла его описывать, так как здесь впринипе все кристально ясно (на вход график -> задаем параметры окна -> выводим график на экран).
   2. averageAgeTeam: метод выполняет 1 задачу. Здесь опять же создается SQL запрос, который мы позже выполняем. Далее я создаю dataset с собранной информацией и заполняю его. Следующая подзадача – вывод информации в виде графика. Я выбрал столбчатый график, который получил значения из dataset.
   3. topFivePlayersFromHighestTeam: выполняет 2 задачу. Создаю SQL запрос, выполняю его и позже записываю данные в переменную result.
   4. Task3 (не придумал как назвать на английском более понятно и в то же время не слишком объемно): выполняет задачу 3. Также создаю SQL запрос и выполняю его, после чего записываю информацию в result.
5. В классе Main я читаю и записываю данные из SportTeams.csv, создаю БД и записываю в неё данные из файла, после чего я выполняю все 3 метода класса Task.

Результат выполнения работы:



