# Глава 2

## 2.1

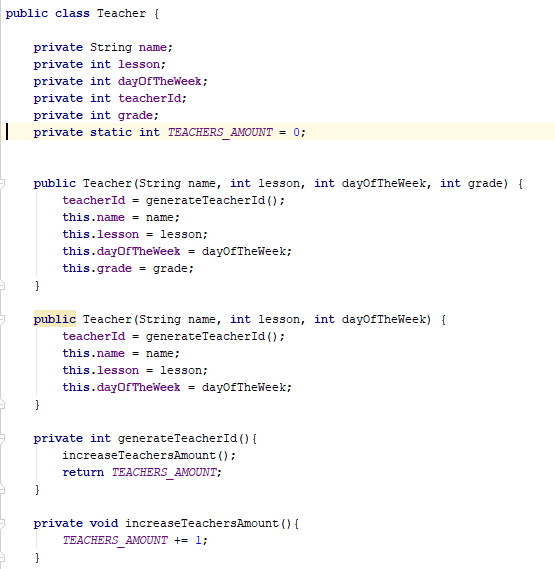
Резюмируя изложенное в главе 1, генетические алгоритмы – мощнейший инструмент для решения задач оптимизации и моделирования. Свое название получил также не случайно, ведь он моделирует генетические процессы, происходящие в живой природе. Исходя из этого мы можем смоделировать основные механизмы (мутация, селекция, скрещивание), наложить необходимые ограничения и эволюционным путем получить итоговое расписание.

Для решения задачи составления расписания используем язык программирования Java 1.8 и среду разработки Intellij Idea Community Edition.

На начальном этапе необходимо определить, из чего алгоритм должен создавать новые особи. Если представить расписание на день в виде одной из тех самых особей, то, для каждого класса, такой день будет состоять из предметов школьной программы. Каждый из предметов ведет определенный учитель. Так как расписание создается исходя из школьной программы и наличия рабочих часов учителей, то за минимальную единицу для составления расписание возьмем преподавателя.

Создадим класс *Teacher* для генерации преподавателей (рисунок 2.1) со следующими полями:

* *name* – имя преподавателя;
* *lesson –* номер урока по порядку;
* *dayOfTheWeek –* день недели, в который проводится урок.
* *teacherId –* уникальный идентификатор объекта класса *Teacher*
* *grade* – номер класса (Например: 1а, 8б)
* *TEACHERS\_AMOUNT –* количество созданных объектов класса *Teacher*.

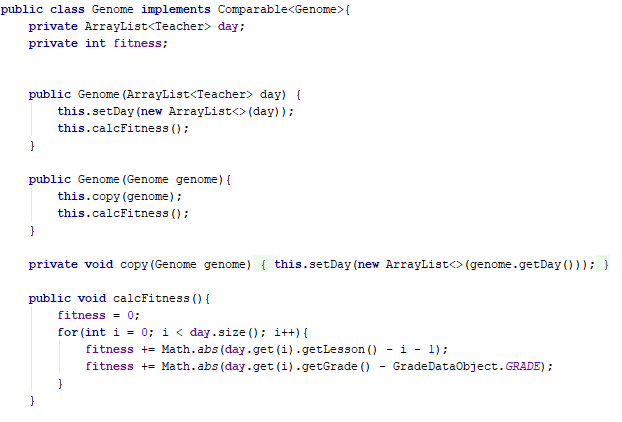


1. - Класс *Teacher*

Для того, чтобы объекты данного класса были действительно атомарными и неизменяемыми, разрешим задавать значение полей только при инициализации. Если учитель проводит несколько уроков в день, то для него создаются объекты в том же количестве (Например: 4 урока – 4 объекта).

Следующим шагом необходимо создать модель особи (генома), которая должна генерироваться с использованием объектов класса *Teacher.* Для этого напишем новый класс *Genome* (рисунок 2.2) со следующими полями:

* *day –* список, содержащий в себе учителей в том порядке, в котором они ведут уроки в конкретный день недели;
* *fitness –* значение функции приспособленности для каждого объекта класса *Genome.*



1. - Класс *Genome*

За вычисление значения функции приспособленности отвечает функция *calcFitness().* Она вычисляет модуль разности между позицией преподавателя в списке *day* и номером урока по порядку, указанный в объекте *Teacher,* и добавляет его к значению переменной *fitness*. Тоже самое происходит и с переменной, которая хранит в себе тот класс, который присутствует на уроке.   
Идеальной особью, которая попадет в финальную популяцию (расписание на день) является та, чье значение *fitness* равно 0.