

## Домашнее задание № 2.

### Марсианские хроники

#### Введение

Система родственных отношений у марсиан довольно оригинальна. Они почкуются когда угодно и как им угодно так, что у марсианина только **один** родитель, а детей может быть и десяток, и сотня, а может и не быть вообще и при этом еще некоторые марсиане могут по желанию менять своих родителей и детей. Марсиане привыкли к этому, и такой жизненный уклад кажется им естественным.

А вот Планетарному Совету запутанная генеалогическая система создает серьезные неудобства. Поэтому по решению Совета была создана Специальная комиссия, которая должна навести порядок раз и навсегда. Был издан Указ, согласно которому:

“Каждая марсианская семья должна предоставить отчет о составе семьи в указанном формате”. (см. ниже)

Данный отчет должен быть подробен настолько, чтобы с помощью марсианских технологий можно было создать полную копию семьи.

Представьте, что вы член Специальной комиссии и вашей задачей является разработка программы, позволяющей создавать такие отчеты. Вам в помощь предлагается информация об особенностях и обязанностях марсиан и методах, которыми вы должны проводить ревизию.

#### Особенности марсиан

Каждый марсианин имеет генетический код, который может быть как String, так и Integer или вообще любым объектом, причем тип этого кода одинаков для всех членов семьи (Marsian<T>). Значение *value* кода у каждого марсианина свое (хотя может и совпадать с кодом какого-нибудь родственника).

Так уж вышло, что марсиане разбиты на две категории: новаторы и консерваторы. Причем они, мягко говоря, не в восторге друг от друга, поэтому вы никогда не встретите новатора и консерватора в одной марсианской семье.

Новаторы любят экспериментировать, поэтому они могут из любопытства поменять себе родителя или забрать к себе чужих детей, сделав их своими и лишив их прежних родителей родительских прав. Могут они и изменить значение генетического кода. В этом обществе такое считается нормой, **главное - случайно не оказаться своим же потомком** (т.е. сыном, внуком и т.д.), **и не выдавать чужих детей за своих**.

У консерваторов считается непозволительным менять свой код и тем более членов своей семьи.

## Обязанности марсиан

Каждый марсианин по требованию Совета обязан:

- предоставить (вернуть Collection) всех своих детей или потомков (детей, детей детей и т.д.)
- указать своего родителя, если его нет, тогда null
- сообщить о наличии у него детей или потомков с указанным значением кода (true или false).

Все новаторы по просьбе Совета обязаны:

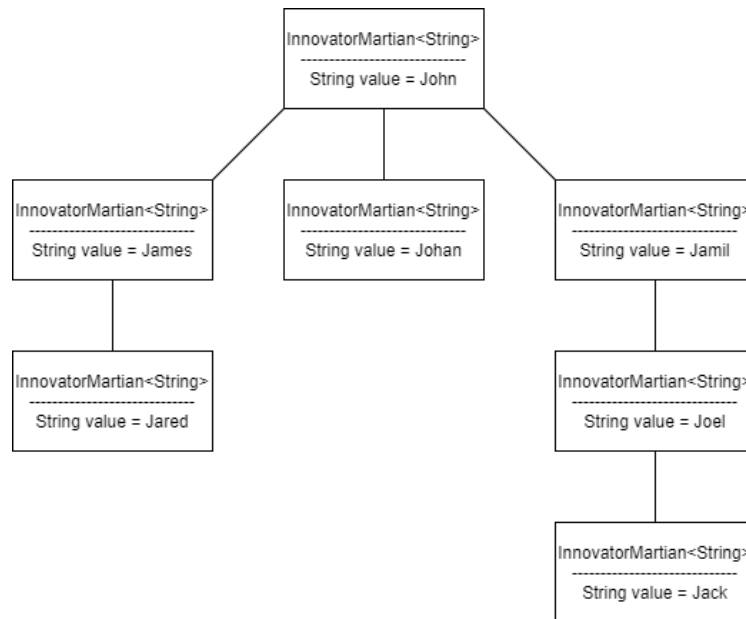
- поменять свое значение генетического кода *value* на указанное
- поменять своего родителя на указанного (причем нужно уведомить об этом текущего родителя)
- поменять своих детей на указанных (прискорбно, но нужно сообщить детям об этом)
- удалить или добавить указанного ребенка. Если это противоречит законам генеалогии, вежливо отказаться (вернуть false)

## Форма отчета

Выводите строковое представление генеалогического дерева в следующем рекурсивно заданном формате: сначала выводится информация о марсианине в формате

**<SimpleRootClassName>(<SimpleObjectClassName>:<ObjectToString>)**,  
где **SimpleRootClassName** имя класса марсианина без указания пакета,  
**SimpleObjectClassName** – имя класса генетического кода марсианина,  
**ObjectToString** – строковое представление кода; затем, в таком же формате, рекурсивно выводится информация о его детях (если они есть) с добавлением лидирующих 4-х пробелов.

Например, для структуры дерева, схематически указанной ниже, будет сгенерировано следующее строковое представление:



InnovatorMartian (String:John)  
InnovatorMartian (String:James)  
InnovatorMartian (String:Jared)  
InnovatorMartian (String:Johan)  
InnovatorMartian (String:Jamil)  
InnovatorMartian (String:Joël)  
InnovatorMartian (String:Jack)

Нужно добавить метод, позволяющий переводить обратно этот отчет в генеалогическое дерево.

В качестве значения **<ObjectToString>** могут выступать: строка (256 знаков максимум), 32-битное знаковое целочисленное число, 64-битное число с плавающей точкой. **<SimpleObjectClassName>** принимает значения: String, Integer, Double соответственно. **<SimpleRootClassName>** может принимать имена ваших классов (новатор и консерватор).

## Требования

1. Все марсиане очень подозрительны. Поэтому, когда они принимают откуда-то или отдают куда-то коллекции детей, они заботятся о том, чтобы никто - ни на словах, ни на деле - не смог изменить эти коллекции без их ведома.
2. Консерваторы, вообще запрещающие что-либо у них менять после их создания, доводят свою подозрительность до предела и требуют, чтобы все поля в описывающих их классах были объявлены **final**.
3. Для всех марсиан надо реализовать:
  - a. getter, который возвращает родителя, а если его нет, тогда null
  - b. getter, который возвращает Collection всех детей, а если их нет, тогда пустую коллекцию
  - c. getter, который возвращает Collection всех потомков, т.е. детей, детей детей и т.д.

- d. метод `hasChildWithValue(T value)`, сообщающий о наличии ребенка с указанным генетическим кодом (`true`, если есть, иначе `false`)
  - e. метод `hasDescadantWithValue(T value)`, сообщающий о наличии потомка с указанным генетическим кодом (`true`, если есть, иначе `false`)
4. Для новаторов надо также реализовать:
- a. setter, который устанавливает новый генетический код.
  - b. setter, который позволяет изменить родителя данного новатора.
  - c. setter, устанавливающий Collection новых потомков новатору
  - d. метод, добавляющий указанного ребенка к данному новатору, если добавление успешно, метод должен вернуть `true`
  - e. метод, удаляющий указанного ребенка у данного новатора, если удаление успешно, метод должен вернуть `true`  
(Два предыдущих метода могут вернуть `false`, если добавление или удаление ребенка нарушает правила марсианской генеалогии, например, если образуется цикл или петля)
5. Реализовать класс генеалогического дерева с методом
- a. предоставляющим это дерево в виде текстового отчета (см. параграф «Форма отчета»)
  - b. переводящим отчет в объект класса генеалогического дерева (см. параграф «Форма отчета»)
6. Консерваторы сразу не рождаются. Они появляются, воспроизводя себя из новаторов, которые решают, что пришла пора сохранить свою неизменяемую копию для вечности. Поэтому нужно реализовать конструктор класса консерватора, который принимает на вход объект класса новатора. Структуры исходного и результирующего деревьев при этом одинаковы, содержат соответствующие одинаковые значения кодов марсиан, но отличаются возможностью (новаторы) и невозможностью (консерваторы) изменения своей структуры.
7. При выполнении любых операций с деревом связи во всех парах родитель-ребенок должны обновляться **корректно**, и граф взаимоотношений марсиан должен всегда оставаться **деревом**.
8. Для реализованных методов написать JUnit-тесты.
9. Проект в Idea должен быть выполнен с использованием JDK 11 и Junit 5

## Результат выполнения задания

Результатом выполнения задания является архив с именем

**HW2\_<номер\_группы>\_<фамилия>\_<имя>.zip**

В нем должен содержаться IntelliJ IDEA проект с выполненным заданием. Лишних файлов и библиотек в архиве быть не должно. Архив загружается в SmartLMS.

**~~Deadline: 23 ноября 2020 г., 23:00 Moscow Time.~~**

**Deadline 26 ноября 2020 г., 23:00 Moscow Time.**