Контролно Java #1

Да се напише програма за симулиране на електрическа схема.

Схемата е съставена от отделни елементи. Всеки елемент има входни(където получава напрежение) и изходни(предава напрежени) връзки, към които могат да се закачат други елементи. Програмата може да изпълни проста симулация на работата на схемата. Симулацията е специална с това, че винаги започва от последния елемент в схемата.

1. **Класове за елементи:**

Всеки елемент има име, което се подава като аргумент на конструктора.
Всеки елемент има списък от входящи и отделен списък от изходящи връзки към други елементи. Всеки елемент има определен брой възможни връзки от всеки вид и те не могат да се надвишават. Връзките реферират директно към другия елемент, който е свързан.
Всеки елемент има метод float eval(), който изпълнява специфичното поведение на елемента и връща неговото изходящо напрежение. Когато поведението изисква напрежение от предишен елемент във веригата се извиква eval на този предишен елемент.

Елемент Generator - това е генератор на напрежение.
- като аргумент на конструктора се подава генерираното напрежение
- има 0 входящи и 1 изходяща връзка
- в eval винаги връща генерираното напрежение

Елемент Lamp - това е краен консуматор, който свети.
- като аргумент на конструктора се подава минималното нужно напрежение за да светне
- има 1 входяща и 0 изходящи връзки
- в eval винаги се взима напрежението от входящата връзка. Ако е по-голямо или равно на нужното лампата светва. Ако е по-малко от нужното лампата не свети

Пример:
gen1 ==> lamp1

Генераторът gen1 има 1 изходяща връзка и тя е свързана към лампата lamp1. lamp1 има 1 входяща връзка и тя идва от gen1. Когато се извика lamp1.eval() ще се вземе генерираното напрежение от gen1 и ако е достатъчно лампата ще светне.

Елемент Splitter - това е междинен елемент, който разделя схемата в няколко посоки
- като аргумент на конструктора получава число N, което определя колко ще са изходните връзки
- има 1 входяща и N изходящи връзки
- в eval винаги се връща стойност V/N, където V е напрежението от входящата връзка и N е броят изходящи връзки

Пример:

/==> lamp1
gen1 ==> split1
\==> lamp2

Ако gen1 генерира 2 волта, те се разделят на 2 части от split1 и всяка лампа получава на входа си по 1 волт.

Елемент Timer - връща изходящо напрежение през определен брой извиквания на eval
- като аргумент на конструктора получава число N, което определя през колко извиквания на eval да връща резултат
- има 1 входяща и 1 изходяща връзка
- в eval се връща стойност 0. На всяко N-то поредно извикване се връща стойността от входната връзка

Пример:
gen1 ==> timer1 ==> lamp1

Ако на таймера се даде N = 3 то lamp1 ще получи напрежението от gen1 на всяко трето извикване на eval.

2. **Общи изисквания:**
    a. В eval на всеки от класовете добавете принтиране на полезна информация, чрез която да може да се проследи изпълнението на симулацията.
    b. Добавете методи за свързване на елементи, които да проверяват дали вече не е превишен броят входящи и изходящи връзки.
    c. В main създайте проста схема, в която да демонстрирате всички елементи.
    d. Ако в дадена входяща връзка няма закачен елемент (има стойност NULL) то входящото напрежение се приема за 0.
    e. На конструкторите на елементи НЕ СЕ подават други елементи, които са свързани към тях. Това свързване да става чрез отделни методи.

3. **Изисквания за 6:**

Елемент Multiplexer - това е елемент, който има множество входове и на базата на контролна стойност определя кой от тях да се прехвърли към изхода
- като аргумент на конструктора получава число N, което определя броя контролни входове
- има общо N + 2^N входящи и 1 изходяща връзка
- първите N входа са контролни, а следващите 2^N входа са нормални. Едновременно се взимат стойностите на всички контролни входове за да се оформи число M в двоичен запис, като за всеки вход напрежение под 0.5 волта се приема за 0, а по-голямо или равно на 0.5 се приема за 1. Така резултатът ще има стойност в интервала [0, 2^N). На изхода се връща стойността от M-тия пореден нормален вход.

Пример:
Ако се добави мултиплексор с N = 2 то той ще има 2 контролни(индекси 0 и 1) и 4 нормални входа(индекси от 2 до 5). При напрежения 0.6 и 0.4 на контролните входове се получава бинарната стойност 10, съответно M = 2. Връща се стойността от входа на индекс 4.

В main имплементирайте и демонстрирайте схемата от прикачената картинка.