

В цифровия свят звуците се представят като последователност от числа. Тези числа се наричат *проби* или *samples*. Стойността на дадена проба отговаря на амплитудата на отместването на механична мембрана в даден момент. Чрез бърза промяна на тази амплитуда се създават звукови вълни и така чуваме звуците, получени от възпроизвеждане на поредица проби. За простота ще приемем, че стойностите на тази амплитуда винаги са числа в определен симетричен затворен интервал $[-M, M]$ и няма да се интересуваме от честотата на семплиране (sampling rate).

В тази задача ще трябва да реализирате полиморфна йерархия от класове, представящи цифрови звуци. Всеки такъв клас ще има две основни характеристики - продължителност (в брой проби) и оператор `[]`, който по подаден номер на пробата връща нейната стойност. Не се предполага промяна на звук след неговото създаване.

Във вашата програма трябва да позволите използване на различни числени типове за описване на пробите, както и различни граници на дефиниционния интервал (константа M). Тези параметри на класа трябва да се подават при създаване на обект.

Реализирайте следните типове звуци:

- *Тишина* - стойността на пробата във всеки момент е 0.
- *Звук от файл* - при конструиране на такъв обект подавате име на файл (произволен символен низ). Този файл ще третирате като двоичен, съставен от последователност от числа, описващи дадения звук. Така например 3-та проба ще е третото число във файла. Приемете, че файлът винаги е с коректен формат.
- *Периодичен звук* - при конструиране се подава масив от числа в типа на звука (*pattern*) и цяло число - колко пъти да се повтори този шаблон.
- *Микс* - при конструиране се подава масив от звуци. Продължителността на звука *Микс* е колкото на най-дългия от тези звуци. Всяка проба представлява сума от съответните проби за всеки от подадените звуци, разделена на броя им. Ако даден звук е по-къс от останалите, то се предполага да се допълни с тишина до нужната дължина.
- *Последователност* - при конструиране се подава масив от звуци. Продължителността на звука *Последователност* е сумата на продължителностите на тези звуци. Полученият звук е конкатенация (долепване) на тези звуци.
- *Ефект* - при конструиране се подава звук и трансформираща операция, която приема като аргумент проба и нейния индекс в звука и връща нова трансформирана проба. Прости примери за ефект е промяна на силата - функцията умножава по даден коефициент всяка проба. Ако коефициентът е по-голям от едно, се получава усилване, ако е по-малък - намаляване. При коефициент 0 се получава ефект Mute. Друг ефект е обръщане на звука - всяка проба се заменя с противоположната ѝ. Така, ако се направи *Микс* от звук и неговия обрънат, ще се получи тишина. Този подход се използва за активно шумоподтискане.

Ако в даден звук се получи стойност на проба, която е извън позволения интервал, тя трябва да се замени с M или $-M$ (в зависимост от знака си). Това е така нареченото *подрязване* (clip).

Конструирайте обект от класа *Ефект*, реализиращ нормализация. Това е ефект, при който даден звук се усилва или намалява автоматично до ниво, така че най-голямата по абсолютна стойност възможна амплитуда да достига подадена граница. По подразбиране тя е M .

Реализирайте клас *Playlist*, който се създава чрез символен низ - име на текстов файл. В този текстов файл всеки ред ще описва звук, който трябва да създадете чрез горните класове. Всеки ред е с не-повече от 1000 символа. Възможни са следните типове редове:

- *Песен* - представлява име на файл (съдържащ песента), следван от число с плаваща точка - коефициент на усилване (2 означава да се изпълни два пъти по-силно, 0 да не се

чува изобщо, 1 да се възпроизведе както е записана и т.н.). Името на песента не съдържа празни символи. Разделител е един интервал;

- *Пауза* - число, което показва продължителността на паузата (тишина) в брой проби.
- *Песен със засилване (FadeIn)*. Представлява символен низ (име на песента) и две числа - първото е брой проби, в които се развива усилването, а второто - крайното ниво. Идеята е да започнете възпроизвеждането от ниво на звука 0 и в продължение на подадения брой проби равномерно да увеличите силата на звука до подаденото като второ число ниво.
- *Преливка* - описанието ѝ представлява две цели числа, последвани от число с плаваща точка. Разделители са един или повече празни символи. Този запис винаги трябва да се намира между две песни и така трите записа създават един звук в резултатния *Playlist*. Целта е при възпроизвеждане предхождащата песен да прелее плавно в следващата. В описанието първото число показва колко проби продължава цялата преливката, второто колко проби от втората песен да прелеят в първата, а третото - какво е нивото на затихване, преди да се включи втората песен. Нека например първото число е А, второто В, а третото С. Ефектът е, че в края на първата песен, силата на възпроизвеждане намалява равномерно в продължение на А-В проби до нивото С, след което в продължение на В проби намалява до 0, като едновременно се включва втората песен, от начало с ниво 0, и в края на първата песен достигне нивото, което е подадено за втората песен. Това също трябва да стане равномерно.

Ваша задача е да валидирате коректността на този файл.

- Класът трябва да поддържа оператор за индексване - по число се връща псевдоним на звука, отговарящ на съответния пореден номер (песен или пауза). Чрез оператора позволете промяна на звука - например подмяната му с друг звук или прилагане на някакъв ефект.
- Реализирайте оператор за преобразуване към псевдоним на звук.
- Също така, трябва да можете да "възпроизведете" целия *Playlist*, като запишете получения от изпълнението му от начало до край звук в съответен двоичен файл.

Демонстрирайте този клас в кратка програма.

Постарайте се решението ви да е ефективно откъм памет (имайте предвид, че звуците може да са дълги).

Помислете как ефектите могат да се постигнат с описаните класове звуци!

Примерен файл за създаване на *Playlist* обект:

```
TheWhiteRabbit.bin 100 1.0
200
AClockIsTickingInMyHome.bin 0.84
200
100
GodIsInTheRain.bin 1.3
1000
LectureNotesInC++.bin 2
HowToOOP.bin 1.7
MyMomSayingToEatMoreVegetables.bin 0
AFriendCallingToPlayFootball.bin 3
200 100 0.3
AFriendInviteMeToABeer.bin 2
```

**При реализация на класовете задължително вземете предвид тяхната използваемост!
Спазвайте ООП парадигмите и добрите практики!**