

## Общи изисквания към решението на задачата:

- Съблюдавайте добрите ООП практики.
- Съветваме ви да използвате само материал, изучаван на лекции / упражнения от курса ООП. Използването на програмни техники и средства извън този материал е на ваша отговорност. За всичко, което не е използвано коректно, ще бъдат отнемани точки.
- Грижете се за коректно използване на паметта.
- Вашата програма трябва да гарантира пълна сигурност при възникване на изключение - не трябва да допускате изтичане на памет, независимо от възникване на някакво изключение, както и получаване на обекти в неконсистентно или некоректно състояние. Всяка операция, която извършвате трябва или да приключва коректно или да остави обектите с които работи в състоянието, в което са били преди започване на операцията.
- Целете максимална гъвкавост и удобство при работа с класовете и обектите.
- Позволено е използването само на функции и класове от следните заглавни файлове на стандартната библиотека на езика C++: `<cassert>`, `<cctype>`, `<cmath>`, `<cstdlib>`, `<cstring>`, `<fstream>`, `<iostream>`, `<new>`, `<stdexcept>`.

## Задача:

В тази задача ще трябва да реализирате абстрактен клас `DataSource`, чиято задача е да бъде източник на данни. Тези данни трябва да могат да бъдат от произволен тип. Всеки елемент от тези данни се извлича от източника и се премахва / преминава се към следващия (т.е. не се извлича един елемент два пъти поред).

Обектите, представители на наследници от този клас трябва да позволяват:

- извличане на един елемент (ако има);
- извличане накуп на определен брой поредни елементи ако има толкова. В случай, че наличните елементи са по-малко да се извлекат колкото са налични;
- проверка дали има следващ елемент. Ако този метод върне истина, то гарантирано следващо извличане на елемент трябва да успее;
- метод `reset`, който възстановява начално състояние на източника (все едно не са извличани елементи и ще се извлекат същите елементи. Внимавайте - това не винаги е възможно! Връща `bool` стойност - дали операцията е била успешна или не;
- оператор `()`, който извлича и връща като резултат един елемент;
- оператор `>>`, който извлича елемент в десния си аргумент. Трябва да позволява слепване (`source >> x >> y`);
- оператор за преобразуване към тип `bool`, който връща истина, ако обектът може да генерира още елементи и лъжа в противен случай.

Преценете какви да са точните прототипи на функциите. Помислете дали ще ви трябват допълнителни методи и ако да - добавете ги към подходящите класове.

Трябва да реализирате следните конкретни наследници:

- Клас `DefaultDataSource`, който може да връща безкрайно много подразбиращо конструирани обекти от типа на данните си.

- Клас `FileDataSource`, който се създава чрез символен низ - име на **текстов** файл. Елементите се четат последователно от този файл.
- Клас `ArrayDataSource`, който се създава чрез масив от елементи и връща последователно тези елементи. За този клас реализирайте също оператори `+` и `+=`, които добавят елемент в края на масива. Реализирайте оператор `-` (префиксен и постфиксен), който връща източника един елемент назад.
- Клас `AlternateDataSource`, който се създава чрез масив от източници на данни и извлича последователно елементи от тях - първо от първия, после от втория, после от третия и т.н. и след извличане на елемент от последния подаден източник отново преминава към елемент от първия и т.н. Ако един източник се изчерпи да се прескача в този ред. Елементите се изчерпват при изчерпване на всички подадени източници.
- Клас `GeneratorDataSource`, който се създава от конструктор с аргумент генератор с поведение на функция, която се използва за генериране на елементите (мислете си например за създаване с генератор `rand` за източник от случайни цели числа без знак).

Да се демонстрира използването на класовете чрез следните две под-задачи:

- Да се създаде източник на низове от малки латински букви. Всеки низ трябва да е с дължина точно 10 символа. Да се изведат на екрана 25 низа извлечени от този източник.
- Да се създаде обект източник на естествени числа, който алтернативно връща поредно просто число, случайно цяло число и поредното от първите 25 числа на Фибоначи (след това те да свършват). Да се запишат 1000 числа от този източник в двоичен файл с подадено от потребителя име. След това да се прехвърлят от този файл в текстов файл, да се създаде файлов източник от него и да се изведат на екрана всички елементи през този файлов източник.