

В тази задача ще реализираме абстрактен клас DataSource, чиято задача е да бъде източник на данни. Тези данни могат да са от произволен тип. Елементите се извличат от източника и предполагаемо се премахват (т.е. не се извличат два пъти поред).

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

- извличане на следващ (пореден) елемент (ако има);
- извличане накуп на определен брой поредни елементи (ако има толкова);
- 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 (напр. 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.

Вие трябва да предоставите следните конкретни наследници:

- Клас DefaultDataSource, който връща безкрайно много, подразбиращо се конструирани елементи от типа на данните си.
- Клас FileDataSource, който се създава чрез символен низ - име на текстов файл и чете елементите от този файл.
- Клас ArrayDataSource, който се създава чрез масив от елементи и връща последователно тези елементи. Реализирайте оператори + и += , които добавят елемент в края на масива. Реализирайте оператор -- , който връща генератора елемент назад.
- Клас AlternateDataSource, който получава като аргументи масив от източници на данни и връща последователно елементи от тях - първо от първия, после от втория, после от третия и т.н. и след последния 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 получава като аргумент на конструктора си генератор, който има поведение на функция, която се използва за генериране на елементите (мислете си за rand).

Да се демонстрира използването на класовете чрез създаване на:

- източник на естествени числа, който алтернативно връща поредно просто число, случайно число и поредното от първите 25 числа на Фибоначи. Числата на Фибоначи да са първите 25, след това свършват. Да се запишат 1000 числа от този източник в двоичен файл с подадено от потребителя име;
- генератор на случайни низове от малки латински букви, всеки с дължина точно 10 символа. Да се запишат 50 низа от този източник в двоичен файл с подадено от потребителя име.