

In tutti gli esercizi non ci si preoccupi delle eccezioni.

Si supponga l'esistenza della classe `java.io.IntegerInputStream` (non presente realmente nella libreria Java).

Esercizio 1

Si scriva il codice di una classe Java **L** che implementi l'entità **lettore da dispositivo**, che legge dei valori interi da un dispositivo di I/O. La classe **L** deve mantenere un riferimento ad un oggetto di classe `java.io.IntegerInputStream`, che rappresenta il dispositivo e che fornisce un metodo `public int read()` che restituisce un intero alla volta.

Nella classe **L** si deve prevedere una operazione di **lettura** che restituisce il valore letto dal dispositivo.

Si preveda un **costruttore** che inizializzi le istanze di **L** con il riferimento ad un oggetto di classe `java.io.IntegerInputStream` fornito.

Esercizio 2

Si scriva il codice di una classe Java **Lcache** che estenda **L**, e che implementi un semplice meccanismo di **cache**. La classe **Lcache** mantiene una cache di **N** elementi implementata come un array di interi. All'inizio, prima di ogni lettura, la cache è *vuota*, cioè non contiene valori validi.

Rispetto alla superclasse si deve **sovrascrivere** l'operazione di lettura per sfruttare la cache:

- se la cache è vuota, si devono leggere **N** valori dal dispositivo di tipo `java.io.IntegerInputStream`, memorizzarli nella cache e restituire il primo valore;
- se invece la cache contiene dei valori validi, deve essere restituito il valore successivo all'ultimo che è stato letto;
- infine, se la cache contiene valori validi e viene restituito l'ultimo valore, si deve svuotare la cache.

Si predispongano opportune variabili di istanza per gestire il meccanismo della cache.

Si prevedano **due costruttori**: uno che inizializzi le istanze di **Lcache** con il riferimento ad un oggetto di classe `java.io.IntegerInputStream` fornito e imposti **N** a 256; l'altro che inizializzi le istanze di **Lcache** con il riferimento ad un

oggetto di classe `java.io.IntegerInputStream` e il valore di **N** forniti come parametri.

Esercizio 3

Scrivere un metodo `main` che usi le classi precedenti e che metta in evidenza almeno un comportamento polimorfico. Si spieghi qual è l'effetto del polimorfismo nel codice scritto.