

Eccezioni

Specifiche

Si definisca una eccezione **TestException** che eredita da **Exception**

Si definisca un componente software che implementa il **main** della applicazione. Il componente riceve **argomenti da linea di comando**.

Il componente definisce un metodo **thrower** che richiede una stringa in ingresso e restituisce un intero.

Specifiche

In particolare:

A) Se la stringa a thrower passata è “**divide**” allora

- 1) Viene definita una variabile intera i ed inizializzata a 0
- 2) Il metodo restituisce i/i

B) Se la stringa a thrower passata è “**null**” allora

- 1) Viene definita una variabile Stringa s ed inizializzata a null
- 2) Il metodo restituisce s.lenght()

C) Se la stringa a thrower passata è “**test**” allora

- 1) Viene sollevata una eccezione TestException
- 2) Il metodo restituisce 0

Specifiche

Il main della applicazione **passa** (ovviamente uno ad uno) al metodo **thrower** tutti gli argomenti ricevuti da linea di comando.

Si richiede che il main gestisca tutte le eccezioni che vengono “eventualmente” sollevate. In particolare le eccezioni gestite sono

- ArithmeticException (vedi API)
- NullPointerException (vedi API)
- TestException

Si richiede che il componente stampi a video il messaggio

- “**Nessuna eccezione è stata sollevata**” nel caso thrower non sollevi eccezioni
- **Il tipo di eccezione sollevata** (es. “TestException”) nel caso contrario

Contatore Circolare

Specifiche

Si definisca una eccezione **MyException** che eredita da **Exception**.

Si definisca un **Contatore Circolare**. Il contatore circolare è caratterizzato dalla base e dal valore attuale del conteggio. Si richiede che il contatore consenta di **resettare** il valore del conteggio e di **settarlo** ad un valore dato. Inoltre il contatore deve permettere di **incrementare** e **decrementare** il valore attuale del conteggio. In particolare nel caso di incremento quando il valore attuale del conteggio ha raggiunto la base, il contatore viene resettato. Nel caso di decremento se il valore del contatore è 0 allora deve essere settato a base-1.

Valore = 3
Base = 5

↑
inc

Valore = 4
Base = 5

↑
inc

Valore = 0
Base = 5

↑
dec

Valore = 4
Base = 5

Specifiche

Si richiede inoltre che la classe che implementa il contatore circolare sollevi una eccezione di tipo **MyException** nel caso in cui venga violata la condizione

$$0 \leq \text{valore del contatore} < \text{base}$$

Questa condizione deve essere testata sia nel **costruttore** dell'oggetto che nel **metodo** che consente il **settaggio** del valore del conteggio.

Specifiche

Si realizzi un componente che software che implementa il **main** della applicazione. Il componente deve:

- **Creare** una istanza della classe che realizza il contatore circolare con **base 6** e valore del **conteggio 5**
- **Creare** una istanza della classe che realizza il contatore circolare con **base 2** e valore del **conteggio 3**
- **Incrementare** il valore del **conteggio** della **prima istanza** creata

Si richiede che vengano **gestite eventuali eccezioni**. In particolare la gestione delle eccezioni deve **stampare a video** il messaggio **“ECCEZIONE!!!”**

Gestore di Contatori Circolari

Esercizio Specifiche

Si definisca una eccezione **MyException** che eredita da **Exception**. La classe è caratterizzata da un attributo intero che registra il numero di istanze della classe MyException che sono state create. Inoltre la classe MyException implementa un metodo che restituisce il valore dell'attributo intero.

Si definisca un **Contatore Circolare**. Il contatore circolare è caratterizzato dalla base e dal valore attuale del conteggio. Si richiede che il contatore consenta di **resettare** il valore del conteggio e di **settarlo** ad un valore dato. Inoltre il contatore deve permettere di **incrementare** e **decrementare** il valore attuale del conteggio. In particolare nel caso di incremento quando il valore attuale del conteggio ha raggiunto la base, il contatore viene resettato. Nel caso di decremento se il valore del contatore è 0 allora deve essere settato a base-1.

Esercizio Specifiche

Si richiede inoltre che la classe che implementa il contatore circolare sollevi una eccezione di tipo **MyException** nel caso in cui venga violata la condizione

$$0 \leq \text{valore del contatore} < \text{base}$$

Questa condizione deve essere testata nel **metodo** che consente il **settaggio** del valore del conteggio. Si richiede che venga settato il messaggio “setup error” nella eccezione.

Esercizio Specifiche

Si implementi il gestore di un array di Contatori Circolari. Si richiede in particolare che il gestore consenta di:

- **Aggiungere** un contatore circolare all'array.
 - qualora ci sia spazio disponibile il contatore viene inserito nell'array, viceversa viene sollevata una eccezione.
- **Eliminare**, un contatore dall'array data la sua posizione
 - Se esiste un contatore nella posizione specificata viene eliminato e l'array viene compattato. Viceversa viene sollevata una eccezione.

Esercizio Specifiche

- **Incrementare/Decrementare/resettare/settare** ad un valore dato un contatore data la sua posizione all'interno dell'array.
 - **Se esiste** un contatore nella posizione specificata viene eseguita l'operazione. **Viceversa** viene sollevata una eccezione di tipo **MyException** con messaggio **"Index Error"**.
- **Stampare a video** il contenuto dell'array gestito

In particolare si noti che nel caso di **settaggio** del valore di un contatore circolare, l'istanza di contatore **potrebbe** sollevare una **eccezione**. In questo caso il gestore **NON** può gestire l'eccezione nel suo contesto di esecuzione e la deve quindi **rilanciare**.

Esercizio Specifiche

Si implementi inoltre un componente software che implementa il **main** della applicazione. Si richiede che il componente Main:

- 1) **istanzi** un gestore di array di contatori circolari
- 2) inserisca **due istanze** di contatore circolare
 - 1) La prima base 3, valore 2
 - 2) La seconda base 5, valore 3
- 3) **incrementi** il valore della **prima** istanza e **stampi a video l'array**
- 4) **setti** il valore della seconda istanza a 7 e **stampi a video l'array**

Esercizio Specifiche

Il componente software deve gestire le eccezioni che eventualmente vengono sollevate. In particolare è richiesto che venga **stampato a video** la **printStackTrace** della eccezione (vedi API). Inoltre si richiede che venga stampato a video il **messaggio di errore della eccezione** e il **numero di volte** che questa è stata sollevata.

Filtro

Esercizio Specifiche

Scrivere un componente software che funga da filtro, ossia che prenda l'ingresso da System.in e ponga la propria uscita sia su System.out sia sul file il cui nome è passato come primo argomento di invocazione. Il filtro deve eliminare tutti i caratteri in ingresso che compaiono nel secondo argomento di invocazione. Se si invocasse con argomenti: `java Filtrofileout.txt stringaMAIUSC%` si dovrebbe eliminare dall'uscita ogni occorrenza dei 14 caratteri della stringa "stringaMAIUSC%".