Gestore di Stringhe

Esercizio – Specifica

Si implementi il gestore di un array di Stringhe.

```
public void add (String st)
```

Consente di aggiungere la stringa st all'array qualora ci sia spazio disponibile, viceversa deve essere stampato a video un messaggio di errore.

```
public void delete (String st)
```

Consente di eliminare dall'array una stringa st specificata. Se la stringa viene trovata l'array viene compattato, viceversa viene segnalato un messaggio di errore.

Esercizio – Specifica

```
public void ordina()
```

Consente di ordinare le stringhe dell'array in ordine alfabetico.

```
public String toString()
```

Restituisce una stringa costituita da un numero di righe pari alla dimensione dell'array. Ogni riga riporta la stringa corrispondente nell'array.

Esercizio – Specifica

```
public void specular (String st)
```

Ricerca all'interno dall'array la stringa st specificata. Se la stringa viene trovata questa viene sostituita dalla stringa speculare ottenuta invertendo l'ordine di tutti caratteri che la compongono, viceversa viene segnalato un messaggio di errore.

GNAM --> MANG

```
public boolean isPalindroma (int indxStr)
```

Restituisce true se la stringa memorizzata nell'array con indice specificato da indxStr è palindroma, false viceversa. Nel caso in cui l'indice specificato da indxStr sia negativo o superiore al riempimento dell'array viene viene stampato a video un messaggio di errore. Le stringhe palindrome possono essere lette in uguale maniera da sinistra a destra e viceversa.

ZEUS SEES SUEZ

Esercizio – Specifica

```
public void makePalindroma (int indxStr)
```

Rende palindroma la stringa memorizzata nell'array con indice specificato da indxStr. Nel caso in cui l'indice specificato da indxStr sia negativo o superiore al riempimento dell'array viene stampato a video un messaggio di errore.

GNAM --> GNAMMANG

ANNA --> ANNA

Esercizio – Specifica

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

1)Deve essere istanziata la classe di **gestione degli array** di stringhe.

2) registrare nel gestore le stringhe “topolino”, “pippo”, “pluto”

3) stampare le stringhe registrate nel gestore

4) **Ordini** le stringhe registrate nel gestore e lo **stampi** nuovamente a video le stringhe registrate nel gestore

5) Rendere speculare la stringa “topolino” e lo **stampare** nuovamente a video le stringhe registrate nel gestore

Ereditarietà

Esercizio – Specifica

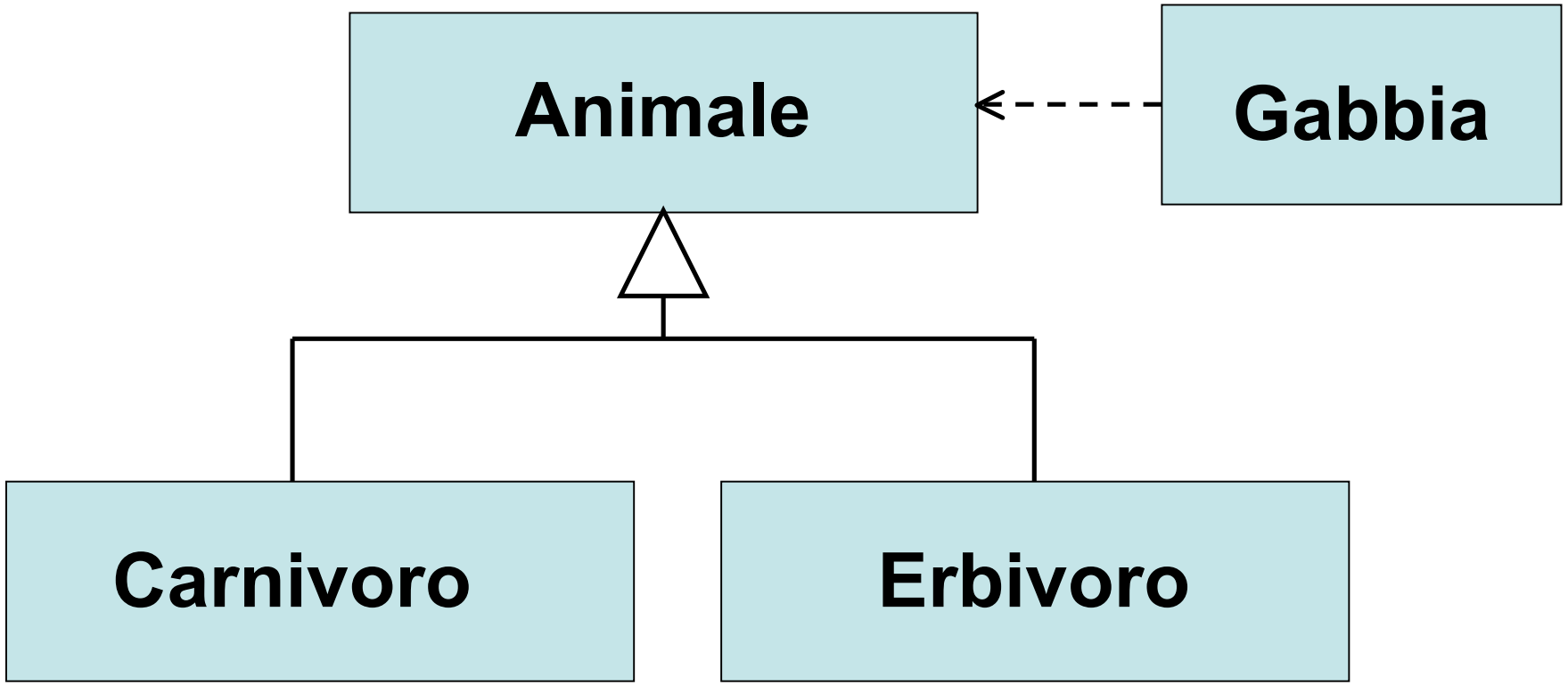
Si progetti ed implementi un componente software che rappresenta la **gabbia** di un zoo.

Ogni **gabbia** è caratterizzata dal **numero massimo** di animali che può contenere e dalla **lista** degli animali che possono essere ospitati.

Tutti gli **animali** sono caratterizzati dal **nome** e da un cibo preferito. In particolare, gli animali possono essere **carnivori** ed **erbivori**. I **carnivori** sono caratterizzati dalla **preda preferita** mentre gli **erbivori** sono caratterizzati dalla **pianta preferita**.

In una stessa gabbia possono essere ospitati sia carnivori che erbivori. In particolare, **non** è possibile che un **animale** ospitato nella gabbia sia il **cibo preferito** (ovvero la preda preferita) di un **carnivoro** ospitato nella **stessa gabbia**.

Esercizio – Specifica



Esercizio – Specifica Gabbia

```
public boolean insAnimale (Animale anim)
```

Consente di inserire un animale nella gabbia. L'inserimento può avvenire solo qualora sia disponibile spazio nella gabbia. Inoltre è richiesto che l'animale specificato da `anim` non sia il cibo preferito di alcun ospite della gabbia e che non abbia come cibo preferito altri animali nella gabbia. Se le condizioni necessarie per l'inserimento sono tutte verificate l'animale specificato da `anim` viene registrato nella gabbia e viene restituito true. Viceversa viene restituito false.

```
public void print ()
```

Stampa a video la lista di tutti gli animali attualmente ospitati nella gabbia.

Esercizio – Specifica Animale

`public String` **`getTipo ()`**

Restituisce la Stringa vuota “”.

`public String` **`getNome ()`**

Restituisce il nome dell’animale

`public String` **`getFood ()`**

Restituisce il piatto preferito dell’animale

`public String` **`toString ()`**

Restituisce una stringa secondo il formato

`Nome [nome animale]`

Esercizio – Specifica Carnivoro

```
public String getTipo ()
```

Restituisce la Stringa “Carnivoro”.

```
public String getFood ()
```

Restituisce il nome del cibo preferito (preda preferita) del carnivoro

```
public String toString ()
```

Restituisce una stringa secondo il formato

`Nome [nome animale], Preda [preda preferita]`

Esercizio – Specifica Erbivoro

```
public String getTipo ()
```

Restituisce la Stringa “Erbivoro”.

```
public String getFood ()
```

Restituisce il nome del cibo preferito (della pianta preferita) dell’erbivoro

```
public String toString ()
```

Restituisce una stringa secondo il formato

`Nome [nome animale], Pianta [pianta preferita]`

Esercizio per Casa
Ereditarietà

Esercizio per Casa – Specifica

Si richiede di progettare un sistema software che simula il comportamento di un numero dato di orologi. Gli **orologi** possono essere **normali** o più specializzati: a **lancette** o **sveglie**.

Ogni orologio è caratterizzato dal ***ore***, ***minuti*** e ***secondi*** che attualmente segna. In particolare, ogni orologio permette di:

- **Settare** l'orario
- Fornire l'orario in formato **stringa** di caratteri
- Controllare l'**equivalenza** con un altro orologio
  - due orologi con stesso orario sono equivalenti
- **Incrementare** l'orario di un **secondo**

Esercizio per Casa – Specifica

Gli orologi hanno funzionalità simili ma non identiche. In particolare, gli orologi differiscono per come rappresentano l'orario.

L'orologio normale e la sveglia restituiscono l'orario in un formato stringa come riportato sotto

[Ore] : [Minuti] : [Secondi]

L'orologio a lancette restituisce l'orario in un formato stringa come riportato sotto

[Ore] AM: [Minuti] : [Secondi]	*Ore <=12*
[Ore-12] PM: [Minuti] : [Secondi]	*Ore > 12*

Esercizio per Casa – Specifica

La sveglia aggiunge una funzionalità in più rispetto agli orologi normali: consente di **settare** un ***allarme*** in termini di ***ore*** e ***minuti***. Ogni volta che viene incrementato un secondo, e l'orario segnato dalla sveglia corrisponde a quello dell'allarme, allora:

1) Viene incrementato il numero dei secondi

2) Viene stampata a video il messaggio riportato sotto:

DRRIIIIIIIIIN!!!! DRIIINNN!!!!

Esercizio per Casa – Specifica

Si **implementi** inoltre un componente software **main**. Il componente software gestisce un array di 3 orologi. In particolare il primo elemento dell'array è un orologio **normale** che segna le **15:13:12**, il secondo è un orologio a **lancette** che segna le **3PM:13:15**, il terzo è una **sveglia** che segna le **15:13:12** e ha l'allarme settato alle **15:18.**

Il componente software deve **incrementare** l'orario di **tutti** gli orologi **60 volte** (60 secondi). **Ad ogni** incremento deve **stampare l'orario** che segnano **tutti** gli orologi.

Traccia di Soluzione