

LABORATORIO DI PROGRAMMAZIONE 2009-2010

PROVA D'ESAME

Andrea Trentini - DICO - UniMi

Luglio 2010

Avvertenza. Non e' ammesso l'uso delle classi del package *prog.io* allegato al libro di testo del corso.

1 Testo esame

Lo scopo e' realizzare una rappresentazione di punti interessanti su una mappa geografica. Tali punti servono a descrivere un luogo interessante turisticamente, hanno eventuali foto associate e link a siti web che parlano del posto.

Le classi da realizzare sono le seguenti:

- *Mappa*, rappresenta la mappa di una zona geografica
- *PuntoDiInteresse*, rappresenta un punto interessante sulla mappa
- *PuntoDiInteresseConFoto*, rappresenta un punto interessante sulla mappa con una o piu' foto associata/e
- *Foto*, rappresenta una foto
- *URL*, rappresenta un sito web
- *Main*, contiene il solo metodo *main()* con alcune invocazioni di test (importante e' riuscire a testare TUTTI i metodi almeno una volta)

Tali classi dovranno esporre almeno i metodi specificati nelle sezioni seguenti. Eventuali metodi di servizio possono essere aggiunti a piacimento.

Ogni classe deve avere il *toString()* che rappresenti lo stato delle istanze e i **costruttori** adeguati per gli attributi che vengono dichiarati.

Si suggerisce, anche dove non segnalato, di utilizzare, se esistenti e se applicabili, le classi parametriche (es. *Vector<E>* invece di *Vector*).

1.1 Mappa

Contiene una lista (usare *Vector<E>*) di punti di interesse (istanze di *PuntoDiInteresse*).

Deve esporre i seguenti metodi **pubblici**:

- *void aggiungiPunto(PuntoDiInteresse num)*, aggiunge un punto alla mappa
- *boolean rimuoviPunto(int indice)*, toglie un punto dalla mappa
- *int numeroPunti()*, restituisce il numero di punti presenti nella mappa
- *int numeroFoto()*, restituisce il numero di foto (associate ai punti) presenti nella mappa

1.2 PuntoDiInteresse

Contiene degli attributi (di tipo opportuno) per memorizzare un punto di interesse geografico. Deve contenere inoltre due attributi per latitudine e longitudine e infine uno per la descrizione testuale.

Oltre ai costruttori opportuni deve esporre i seguenti metodi **pubblici**:

- *void setLatitudine(int num)* (numeri positivi = lat nord, in secondi)
- *void setLongitudine(int num)* (numeri positivi = long est, in secondi)
- *boolean setDescrizione(String testo)*
- *String getDescrizione()*

1.3 PuntoDiInteresseConFoto

Estende *PuntoDiInteresse* aggiungendo la parte di gestione delle foto associate.

Oltre ai costruttori opportuni deve esporre i seguenti metodi **pubblici**:

- *void associaFoto(Foto foto)*, aggancia una foto a questa segnalazione (possono esserci piu' foto associate allo stesso punto)
- *int quanteFoto()*, restituisce il numero di foto agganciate a questa segnalazione

1.4 Foto

Deve contenere gli attributi (di tipo opportuno) per rappresentare la larghezza e l'altezza dell'immagine.

1.5 URL

Contiene degli attributi (di tipo opportuno) per memorizzare un url di un sito.

Oltre agli opportuni costruttori deve esporre i seguenti metodi **pubblici**:

- *void setUrl(String url)*, permette di impostare l'URL, non e' necessario fare complicati controlli di coerenza, ma il metodo deve verificare che la stringa inizi con "http://"
- *String getUrl()*

1.6 Main

Deve contenere il metodo *main()* in cui vanno istanziate le classi realizzate e opportunamente testate, ad es. creando almeno una mappa e popolandola di alcuni punti di interesse, etc.

2 Consegna

Ricordo che le classi devono essere tutte *public* e che vanno consegnati tutti i file *.java* prodotti. NON vanno consegnati i *.class*

Per la consegna, eseguite l'upload dei singoli file sorgente dalla pagina web: <http://upload.dico.unimi.it>.