

progetti di esame

a.a. 2018/2019

Modalità di esame

Le modalità dell'esame orale e dei relativi progetti seguono queste linee:

- Il progetto va svolto o singolarmente o in gruppo di al più 2 persone.
- Il progetto deve essere consegnato almeno 3 giorni prima dell'esame scritto; è ammissibile ma sconsigliato che la consegna sia parziale, a patto che contenga perlomeno il 90% del codice.
- La consegna del progetto deve essere accompagnato da una **breve relazione** (massimo una facciata) in cui vengono illustrate la struttura delle classi e le scelte adottate.
- Nella valutazione del progetto rientrerà la conformità dell'implementazione scelta con il paradigma di programmazione a oggetti, e la conformità del codice presentato alle regole di buona programmazione e convenzioni Java.
- La discussione orale del progetto è individuale, a turno tra gli elementi del gruppo, e verte sulle modalità di implementazione adottate e sulla padronanza dei concetti necessari per realizzare il progetto (spiegati a lezione).
- La valutazione dell'esame orale sarà quindi composta dalla valutazione del progetto (per gruppo) e dalla valutazione della padronanza mostrata nel colloquio (singolarmente).

Note generali

Tutti i progetti presentati devono seguire questa struttura e requisiti:

- Netta e totale separazione tra le classi relative al "modello" che risolve il problema affrontato e quelle relative all'interfaccia con l'utente.
- Per quanto riguarda le classi del modello, devono sempre:
 - *avere un compito preciso e fare soltanto quello;*
 - *non contenere alcuna lettura/scrittura da console o altra interazione con l'utente;*
 - *lanciare eccezioni dove appropriato;*
 - *essere dotate di classi di test;*
 - *essere documentate con javadoc (almeno per i componenti pubblici).*
- Per quanto riguarda le classi dell'interfaccia utente, possono a scelta realizzare:
 - *un interfaccia grafica, che allora deve essere strutturata secondo l'architettura MVC;*
 - *un interfaccia testuale (che sostituisce entrambe le componenti "vista" e "controllo").*
- Cercate sempre di relazionare le classi tra loro ovunque applicabile, facendo uso di ereditarietà, classi astratte e interfacce dove necessario.
- **Cercate sempre di ragionare su quali componenti di una classe dovrebbero essere pubblici e quali privati/protetti, quali statici e quali di istanza, quali costanti e quali no.**
- Cercate di mantenere la dimensione dei metodi contenuta, spezzandoli ove appropriato, e rispettare le convenzioni di scrittura codice Java.
- È sempre *fortemente* sconsigliato inserire nei progetti elementi fuori dal programma del corso, come database, collegamenti online, etc.

gestione personaggi

Progettare ed implementare in Java un sistema per la gestione di un **personaggio di un gioco di ruolo**. Il sistema deve comprendere *almeno*:

- Punteggi, che hanno un *valore* numerico, una *descrizione* testuale, e possono essere incrementati pagando un certo *costo* (diverso per ogni punteggio).
- Qualità, caratterizzate da un *costo*, dei *requisiti* necessari per acquisire la qualità, una *descrizione* testuale, e un *effetto*.
- Requisiti, che possono essere:
 - *un valore minimo per un punteggio;*
 - *il possesso di una data qualità.*
- Effetti, che possono essere:
 - *una modifica a un punteggio;*
 - *una modifica del costo di un punteggio.*
- Personaggi, che raccolgano punteggi e qualità, il costo totale pagato finora, e che devono *almeno*:
 - *poter essere memorizzati e letti da file (in formato testuale o binario);*
 - *consentire di aumentare punteggi, registrandone il costo;*
 - *elencare le qualità di cui si posseggono i requisiti;*
 - *consentire di aggiungere una qualità di cui si posseggono i requisiti.*
- Regolamento, che raccoglie tutti i punteggi e qualità che esistono nel gioco.
- Un interfaccia testuale o grafica, che fornisca a un utente un accesso alle funzionalità descritte.

Non ci sono restrizioni all'aggiunta di ulteriori funzionalità pertinenti a vostro piacimento, sempre evitando di uscire dai contenuti del corso.

pac man

Progettare ed implementare in Java una versione semplificata del classico gioco *Pac Man*:

<http://www.freepacman.org>

L'implementazione dovrà comprendere *almeno*:

- Celle per guidare la collisione tra oggetti, contenenti *almeno*:
 - *l'elenco degli oggetti grafici presenti nella cella;*
 - *l'elenco delle celle adiacenti.*
- Oggetti grafici, strutturati in una apposita gerarchia di classi, tra cui:
 - *i muri, attraverso cui non si può passare;*
 - *le pillole che pac-man mangia, e gli danno punti;*
 - *il pac-man, che l'utente può muovere;*
 - *i fantasmi che danno la caccia a pac-man, e gli tolgono una vita se lo raggiungono.*
- Un Arena di gioco, comprendente *almeno*:
 - *il punteggio e il numero di vite del giocatore;*
 - *la griglia bidimensionale delle celle contenute nella configurazione;*
 - *un metodo per caricare la configurazione iniziale da file oppure per generarla casualmente (scegliete una delle due possibilità);*
 - *un metodo per far evolvere l'arena di un unità di tempo, spostando gli oggetti dinamici e accorgendosi se il giocatore vince o perde.*
- A scelta, potete prevedere diversi tipi di pillole (che danno bonus speciali al pac man), oppure diversi tipi di fantasmi (che inseguono il pac man secondo strategie diverse).
- Un interfaccia grafica, che consenta all'utente di giocare una partita visualizzandone lo stato.

Non ci sono restrizioni all'aggiunta di ulteriori funzionalità pertinenti a vostro piacimento, sempre evitando di uscire dai contenuti del corso.