

Cognome

Nome

matricola

Note:

- Tutti i diagrammi disegnati devono utilizzare la sintassi del linguaggio UML 2.x ed essere opportunamente commentati.
- Il foglio protocollo va utilizzato solo per la brutta copia: per le risposte, utilizzare le parti bianche di ogni foglio, in modo tale che ogni risposta stia sul foglio (fronte e retro) corrispondente alla domanda.
- Per i fogli di consegna usare la penna e non la matita.
- Riportare il proprio nome, cognome, matricola, nell'intestazione di tutti i fogli consegnati.

Punti 6/30

Dzone è un *technical social blog*. Al suo interno, gli utenti registrati possono scrivere piccoli articoli (*post*), che riguardano il mondo dello sviluppo e della programmazione. Il sito fornisce a tutti gli utenti una pagina principale nella quale vengono riportati i titoli degli articoli più popolari del momento. È possibile visualizzare anche gli articoli pubblicati semplicemente nelle ultime 24 ore e negli ultimi 7 giorni. Selezionando uno dei titoli, l'utente può visualizzare la pagina di dettaglio dell'articolo, che può essere sia una pagina interna a Dzone, sia una pagina di un *blog* esterno, che viene aperta come sotto-pagina di Dzone. In entrambi i casi, un utente registrato, può votare un articolo per farlo salire nella graduatoria, o può fornire un *feedback* negativo. Gli utenti registrati possono lasciare commenti sugli articoli interni a Dzone. Un commento può iniziare un nuovo *thread* o può essere in risposta ad un precedente commento. Chiunque può condividere l'articolo su Twitter. La registrazione di un utente a Dzone avviene inserendo la propria email (univoca all'interno del sistema) e decidendo una *password*. Il proprio profilo prevede un'immagine, un *nickname* e un collegamento al proprio account Twitter e Facebook. Ad ogni utente è associato un distintivo sulla base del numero di articoli che ha scritto. Al raggiungimento del decimo articolo, ad esempio, si raggiunge il distintivo di "editore". Gli articoli sono suddivisi per argomenti, ad esempio Java, BigData, Mobile, NoSQL. Qualsiasi utente può ricercare un articolo per titolo o può semplicemente navigare le varie categorie messe a disposizione dal sistema.

Si utilizzino i diagrammi dei casi d'uso per modellare gli scenari descritti. Non è richiesta la descrizione testuale dei casi d'uso individuati.

Cognome

Nome

matricola

Punti 7/30

Google sta sviluppando in gran segreto un nuovo linguaggio di programmazione, Gle. L'utente interagisce direttamente con un programma testuale avviabile da shell *gle*>. Una volta avviato *gle* è possibile fornirgli alcuni comandi, come ad esempio il comando di compilazione di un *file* di estensione *.gle*, il che si ottiene utilizzando l'opzione "-c". La compilazione avviene in più fasi. Inizialmente il *file* viene letto e un modulo di *parsing* trasforma il testo in una struttura ad albero di ricerca. Successivamente, il compilatore vero e proprio genera il codice eseguibile a partire dalla struttura ad albero. Inizialmente, Google ha deciso di appoggiarsi sulla JVM per l'esecuzione dei programmi scritti in Gle. Il risultato della compilazione, quindi, è un *file* contenente *bytecode*, eseguibile successivamente su una normale JVM. Poiché il linguaggio è agli inizi la sua sintassi e le funzionalità offerte non sono ancora stabili, il modulo di *parsing* e il compilatore, vengono forniti in varie versioni. Dalla linea di comando è possibile selezionare la versione del linguaggio con la quale si intende compilare del codice sorgente, utilizzando l'opzione "-v n" dove 'n' è il numero di versione. Nel caso non venga specificata alcuna versione, verrà utilizzata l'ultima disponibile. Compilatore e *parser* vengono utilizzati sempre con versioni omogenee tra loro. Una volta prodotto il *bytecode*, è possibile eseguire il programma, utilizzando sempre la shell *gle*>. Il modulo di esecuzione utilizza il programma '*java*' della JVM, adattandone opportunamente alcune sue funzionalità.

Considerando il programma *gle*> come una interfaccia utente basata sul pattern MVC, si modelli il sistema descritto utilizzando un diagramma delle classi e gli opportuni *design pattern*. Inoltre, utilizzando un diagramma di sequenza, si descriva l'intera operazione di compilazione.

Cognome

Nome

matricola

Punti 3/30

Riportare i diagrammi delle classi che caratterizzano le varianti del *pattern* Adapter, ossia Class Adapter e Object Adapter e completare le seguenti affermazioni inserendo la corretta variante del *pattern* negli appositi spazi.

- Il *pattern* _____ permette di adattare più tipi (*Adaptee* e le sue sottoclassi)
- Il *pattern* _____ permette di modificare alcune caratteristiche dell'*Adaptee*.

Cognome

Nome

matricola

Punti 5/30

Proporre concisamente una strategia di verifica tramite *test* sostenibile all'interno di un impegno di taglia analoga al progetto didattico, per dimensioni, livello di qualità atteso, disponibilità ed esperienza delle risorse. Concentrare la risposta sul tipo di test da prevedere, i loro obiettivi, il grado di automazione, e ogni altro fattore ritenuto di interesse. Corredare la risposta motivando ogni elemento della proposta.

Punti 5/30

Fissando l'attenzione sulla definizione di processo associata allo standard ISO/IEC 12207, indicare quali processi sia possibile e opportuno istanziare su una attività di tagli analoga al progetto didattico, e con quale istanziazione concreta (cioè verso quale insieme di attività, obiettivi, e flussi di dipendenza).

Punti 4/30

Spiegare concisamente la differenza tra il modello di sviluppo “iterativo” e quello “incrementale”, fornendo sufficienti elementi per comprendere sia l'uno che l'altro. Nella spiegazione, descrivere anche le condizioni al contorno che rendano preferibile l'uno o l'altro.