Dipartimento di Informatica Sistemistica e Comunicazione
Corso di Laurea in Informatica

Programmazione 2

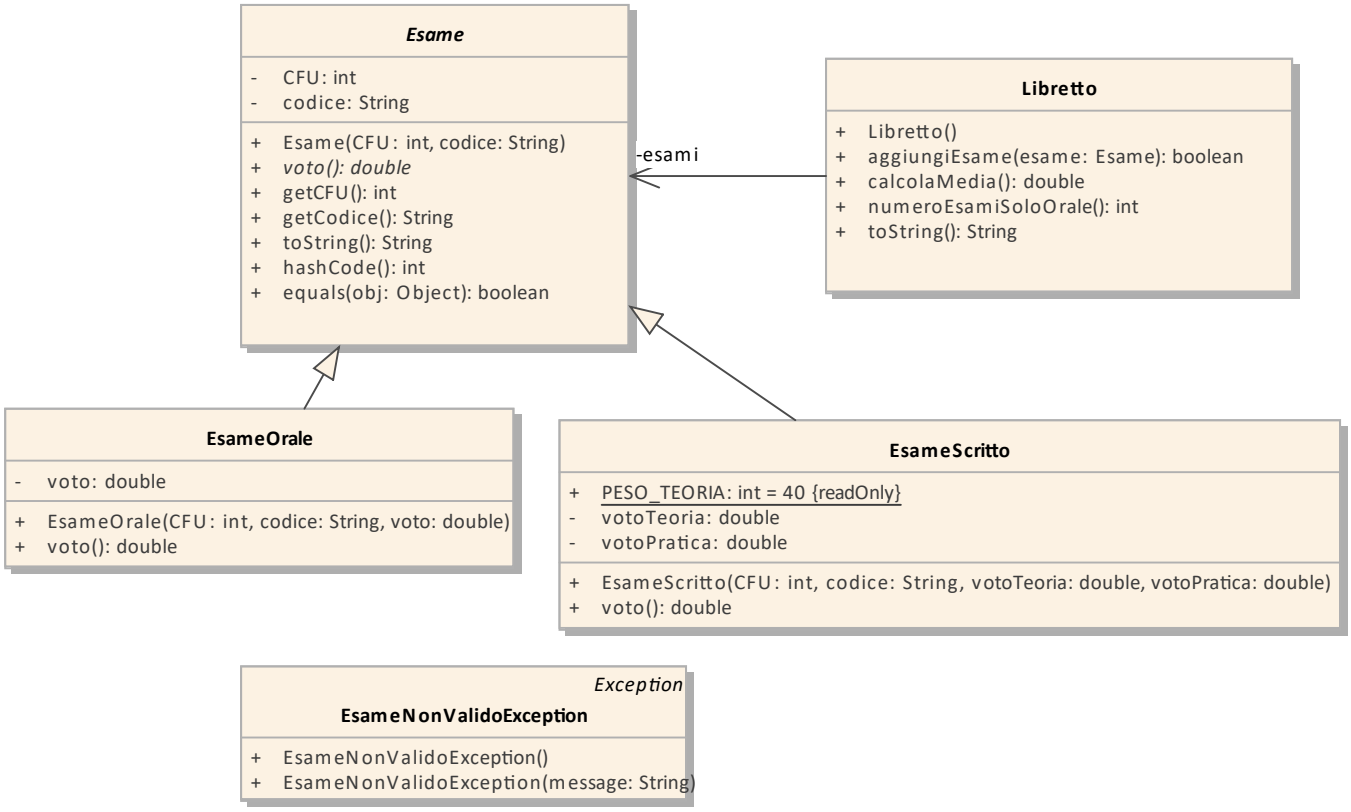
22 Luglio 2021 – Recupero Secondo Compitino

Testo parte di pratica

Si consideri un programma che gestisce la registrazione degli `Esami` universitari sul `Libretto`. Gli Esami possono essere o Orali (caratterizzati da una sola prova con voto) o Scritti (includono la verifica sia della parte teorica e sia di quella pratica). Il voto finale di un esame Scritto è dato dalla media pesata dei voti ottenuti nella parte teorica e in quella pratica.

Implementare le classi esattamente come rappresentate dal seguente diagramma UML. **Il diagramma include tutti e i soli metodi richiesti, compresi quelli di incapsulamento.**

Viene fornita la classe **`TestEsame`** (non mostrata nel diagramma) che contiene un insieme di casi di test che devono essere fatti girare di volta in volta in modo da verificare la corretta realizzazione del programma.



Classe `EsameNonValidoException`:

✓ Rappresenta una eccezione che può essere sollevata dal programma in determinate occasioni.

Classi `Esame, EsameOrale e EsameScritto`:

✓ Rappresentano una gerarchia di 3 classi per la rappresentazione degli esami

Classe **`Esame`**

✓ `Esame` è una classe *astratta* con 2 attributi: `CFU` e `codice`. Il primo rappresenta il peso dell'esame in crediti, mentre il secondo rappresenta un codice univoco che identifica l'esame

- ✓ Il costruttore `Esame(int CFU, String codice)` inizializza i 2 attributi sollevando un'eccezione di tipo `EsameNonValidoException` qualora il parametro `CFU` assuma un valore minore o uguale a 0, oppure `codice` sia `null` o la stringa vuota.
- ✓ Il metodo `voto()` è un metodo astratto che calcola e restituisce il voto dell'esame
- ✓ Il metodo `toString()` ritorna una stringa con codice, CFU e voto dell'esame
- ✓ Il metodo `equals()` confronta due esami e ritorna `true` se sono dello stesso tipo e hanno lo stesso codice
- ✓ Il metodo `hashCode()` ritorna un hash identificativo dell'esame consistente con il metodo equals
- ✓ I metodi `getCFU()` e `getCodice()` ritornano il valore dell'attributo `CFU` e dell'attributo `codice`, rispettivamente

Classe **`EsameOrale`**

- ✓ Estende la classe `Esame` aggiungendo un attributo `voto` che rappresenta il voto dell'esame
- ✓ Il costruttore permette di inizializzare gli attributi definiti nella classe `Esame` e l'attributo `voto`. Se il valore del parametro `voto` è <18 o >31, il costruttore solleva un'eccezione di tipo `EsameNonValidoException`
- ✓ Il metodo `voto()` ritorna il voto relativo all'esame

Classe **`EsameScritto`**

- ✓ Estende la classe `Esame` aggiungendo una costante `PESO_TEORIA` che rappresenta il peso percentuale del voto di teoria sul voto complessivo nonché gli attributi `votoTeoria` e `votoPratica` che rappresentano rispettivamente i voti della parte teorica e pratica dell'esame.
- ✓ Alla costante `PESO_TEORIA` va assegnato valore 40: si intende che la teoria pesa per il 40% sul voto complessivo
- ✓ Il costruttore permette di inizializzare gli attributi definiti nella classe `Esame` e gli attributi `votoTeoria` e `votoPratica`. Se il voto complessivo dell'esame risulta essere <18 o >31, il costruttore solleva un'eccezione di tipo `EsameNonValidoException`
- ✓ Il metodo `voto()` ritorna il voto complessivo dell'esame pesandone le due componenti (teoria e pratica) in base al peso descritto da `PESO_TEORIA`

*(peso_teoria*votoTeoria + (100- peso_teoria)*votoPratica)/100*

Classe Libretto:

- ✓ Ha un attributo che memorizza un insieme di esami. Il libretto non ammette due esami uguali. La collezione utilizzata deve quindi essere un `HashSet`
- ✓ Il costruttore di default inizializza opportunamente la collezione
- ✓ Il metodo `aggiungiEsame(Esame esame)` aggiunge l'esame `esame` alla collezione se diverso da `null`, altrimenti solleva una eccezione di tipo `EsameNonValidoException`. Il metodo ritorna `true` se l'esame viene effettivamente aggiunto alla collezione, `false` altrimenti
- ✓ Il metodo `calcolaMedia()` ritorna la media pesata degli esami in base al rispettivo numero di CFU. La media pesata si calcola come segue:
    1. Moltiplicare il voto di ogni singolo esame per il numero di crediti corrispondenti
    2. Sommare tutti i risultati
    3. Dividere il valore ottenuto per la somma totale del numero di crediti
- ✓ Il metodo `numeroEsamiSoloOrale()` ritorna il numero di esami solo orali presenti nella collezione
- ✓ Il metodo `toString()` ritorna la descrizione (CFU, codice e voto) di tutti gli esami presenti nel libretto