

Programmazione ad Oggetti – traccia B

Appello del 21 Marzo 2024

Punteggi

Task 1 (T1): 6 punti

Task 2 (T2): 6 punti

Task 3 (T3): 6 punti

Aderenza alla traccia (A): 2 punti

Rispetto del paradigma ad oggetti (R): 2 punti

Organizzazione del codice (O): 2 punti

Sufficienza: 12 punti

Punteggio massimo: 24 punti

Termine ultimo consegna: 18/03/2024 23:59

A-L: consegna inviando lo zip del codice a e.corradini@univpm.it

M-Z: consegna inviando lo zip del codice a s.raggiunto@univpm.it

Nella consegna di gruppo, specificare il numero del gruppo ed i componenti.

Contesto: Next Game è una rinomata software house specializzata nello sviluppo di videogiochi innovativi per piattaforme mobili e desktop. La società è in costante crescita e gestisce diversi progetti simultaneamente, ognuno con il suo team di sviluppatori dedicato. L'obiettivo è consegnare prodotti di alta qualità entro i tempi e i budget stabiliti, garantendo al contempo la soddisfazione del cliente e la crescita aziendale.

Obiettivo del Progetto: Sviluppare un sistema software per la gestione dei progetti alla Next Game. Il sistema dovrà consentire il monitoraggio dello stato dei progetti, l'allocazione delle risorse umane e materiali, la pianificazione delle attività e la valutazione delle prestazioni del team.

Componenti da Sviluppare:

1. **Classe Astratta Pianificazione:**
 - Classe astratta dalla quale vengono ereditate le classi Progetto e Task.
 - Attributi: **nome, descrizione, data di inizio, data di scadenza, stato.**
 - Contiene i metodi astratti per la gestione dei progetti e dei task.
2. **Classe Progetto:**
 - Gestisce le informazioni di un singolo progetto.
 - Metodi per la gestione delle informazioni sul progetto.
3. **Classe Sviluppatore:**
 - Rappresenta uno sviluppatore che lavora su uno o più progetti.
 - Attributi: **nome, cognome, competenze, task assegnati.**
 - Metodi per la gestione delle informazioni sullo sviluppatore.
4. **Classe Task:**
 - Rappresenta un'attività specifica da completare all'interno di un progetto.
 - Attributi aggiuntivi: **progetto associato.**
 - Metodi per la gestione delle informazioni sul task.
5. **Classe GestoreProgetti:**
 - Coordina l'intera gestione dei progetti.
 - Contiene la lista dei progetti attivi e dei membri del team.
 - Metodi per il monitoraggio dello stato dei progetti, l'allocazione delle risorse e la valutazione delle prestazioni del team.
6. **Interfaccia DataReader**
 - Definisce i metodi per la lettura dei dati dei progetti da un file di testo.
 - Definisce un metodo generico leggiPianificazione() per la lettura dei dati dei progetti e dei task.
7. **Classe FileReader (Implementazione di DataReader):**
 - Responsabile per la lettura e l'interpretazione dei dati dei progetti da un file di testo.
 - Implementa i metodi definiti nell'interfaccia DataReader per leggere le informazioni sui progetti, gli sviluppatori e i task, convertendoli in oggetti utilizzabili dal sistema.

Requisiti Aggiuntivi: Il sistema dovrà gestire situazioni impreviste come modifiche dei requisiti del progetto, ritardi nello sviluppo e risorse umane limitate. Si richiede un'attenzione particolare alla progettazione di un codice robusto e flessibile, in grado di gestire efficacemente queste eventualità.

File dei Dati: Struttura e Formato

Il file dei dati contiene tutte le informazioni necessarie per la gestione dei progetti alla Software House Next Game. Ecco la struttura e il formato del file:

1. Formato Generale:

- Il file è suddiviso in tre sezioni principali: Progetti, Sviluppatori, Task
- Ogni sezione è separata da un segno di separazione **###** per facilitarne l'identificazione.

2. Sezione Progetti:

- Contiene informazioni sui progetti attivi della Next Game.
- Formato di ciascuna riga: **Nome_Progetto, Descrizione_Progetto, Data_Inizio, Data_Scadenza, Stato_Progetto**.
- Esempio di righe:
Progetto_RocketLeague, Descrizione del Progetto Rocket League, 2024-02-01, 2024-05-31, In Corso
Progetto_SuperMario, Descrizione del Progetto Super Mario, 2024-03-15, 2024-08-30, In Attesa

3. Separatore di Sezione:

- Una riga con **###** che segnala la fine della sezione Progetti e l'inizio della sezione Sviluppatori.

4. Sezione Sviluppatori:

- Elenco degli sviluppatori che lavorano alla Next Game.
- Ogni riga descrive uno sviluppatore con il formato: **Nome_Sviluppatore, Cognome_Sviluppatore, Competenze_Sviluppatore, Task_Associati**.
- Esempio di righe:
Mario, Rossi, Unity, Task_ImplementazioneFunzionalità, Task_OttimizzazionePerformance, Task_Multiplayer, Task_CityDesign, Task_HeistDesign
Luigi, Verdi, Unreal_Engine, Competenze3DModeling, Task_BossFight, Task_LevelDesign

5. Separatore di Sezione:

- Una riga **###** per indicare la fine della sezione Sviluppatori e l'inizio della sezione Task.

6. Sezione Task:

- Informazioni sui task da completare all'interno dei progetti.
- Ogni riga contiene i dettagli di un task con il formato: **Nome_Task, Descrizione_Task, Data_Inizio_Task, Data_Scadenza_Task, Stato_Task, Progetto_Associato**.
- Esempio di righe:
Task_ImplementazioneFunzionalità, Implementazione della nuova funzionalità, 2024-02-15, 2024-03-15, In Corso, Progetto_RocketLeague
Task_OttimizzazionePerformance, Ottimizzazione delle prestazioni del sistema, 2024-04-01, 2024-05-01, Da Iniziare, Progetto_RocketLeague

Istruzioni per la Lettura dei Dati: La classe **FileDataReader** dovrà essere in grado di interpretare questa struttura, leggendo e convertendo le informazioni di ogni sezione in oggetti utilizzabili dal sistema. È fondamentale che la classe gestisca correttamente la distinzione tra le sezioni e che estragga accuratamente i dati in conformità con il formato previsto.

Task 1: Report Progetto

- **Obiettivo:** Generare un report che presenti le informazioni sui progetti della Next Game, inclusi il numero di sviluppatori coinvolti, il numero di task richiesti e l'elenco degli sviluppatori associati a ciascun task
- **Output:** *Nome_Progetto: Numero_Sviluppatori_Coinvolti, Numero_Task_Richiesti, [Nome_Sviluppatore1, Nome_Task1], [Nome_Sviluppatore2, Nome_Task2], ...*
- **Descrizione:** Questo task fornisce un report dettagliato sui progetti, mostrando il numero di sviluppatori coinvolti, il numero di task richiesti e l'elenco degli sviluppatori associati a ciascun task. Per ogni progetto, il report include:
 - Il nome del progetto.
 - Il numero totale di sviluppatori coinvolti nel progetto.
 - Il numero totale di task richiesti per il progetto.
 - Per ciascun task del progetto, viene mostrato il nome dello sviluppatore associato e il nome del task. Se più sviluppatori sono associati allo stesso task, vengono elencati insieme.

Task 2: Analisi delle Deadline dei Progetti

- **Obiettivo:** Analizzare le date di inizio e scadenza dei progetti e dei task per individuare il primo e l'ultimo task di ciascun progetto.
- **Output:** *Nome_Progetto: Nome_Primo_Task, Nome_Ultimo_Task*
- **Descrizione:** Questo task analizza le date di inizio e scadenza dei progetti e dei task per individuare il primo e l'ultimo task di ciascun progetto. Per ogni progetto, vengono identificati il nome del primo task (quello con la data di inizio più anticipata) e il nome dell'ultimo task (quello con la data di scadenza più tardiva). Se sono presenti Task paralleli, creare un ArrayList.

Task 3: Monitoraggio dello Stato dei Task

- **Obiettivo:** Identificare i progetti attualmente in corso e tenere traccia dello stato di ciascun task all'interno di questi progetti.
- **Output:** *Nome_Progetto_InCorso: [Nome_Task, Stato_Task]*
- **Descrizione:** Identificare i progetti attualmente in corso analizzando lo stato di ciascun progetto. Per ogni progetto in corso, individuare tutti i task associati e tenere traccia dello stato di ciascun task. Generare un output che mostri il nome del progetto in corso seguito da un elenco di task associati a tale progetto, insieme al loro stato attuale.