UniLib – Documento di Progetto

Programmazione Avanzata – UNIPI

Niccolò Scatena (518402)

Responsabilità delle classi

UniLib: inizializza la GUI; inizializza le aree (**AreaLibretto**, **AreaModifica**, **AreaStatistiche**); permette il coordinamento degli eventi tra le varie aree; utilizza **EventoDiNavigazione** per inviare eventi a **ServerLogger**; ottiene dalla configurazione l'istanza della classe **Studente** relativa allo studente che sta utilizzando l'applicazione.

AreaLibretto: inizializza la GUI relativa all'area centrale; inizializza la **TabellaEsami** con gli esami dello studente che sta utilizzando l'applicazione; risponde agli eventi di selezione degli elementi in **TabellaEsami** e di pressione del tasto "ELIMINA"; utilizza la classe **Studente** per rimuovere esami dall'archivio; offre metodi *wrapper* per (de)selezionare un esame in **TabellaEsami** e ottenere l'esame selezionato.

AreaModifica: inizializza la GUI relativa all'area di destra; risponde agli eventi di pressione del tasti "SALVA" e "ANNULLA"; utilizza la classe **Studente** per aggiungere e modificare esami dall'archivio; offre metodi per ottenere i valori inseiriti in input dall'utente.

AreaStatistiche: inizializza la GUI relativa all'area di sinistra; inizializza il **GraficoVoti** con la media dei voti per ogni esame nell'archivio; utilizza la classe **Statistiche** per ottenere le statistiche sullo studente che sta utilizzando l'applicazione.

TabellaEsami: mostra in tabella l'elenco degli esami sostenuti dallo studente che sta utilizzando l'applicazione.

GraficoVoti: mostra in un grafico a barre la media dei voti per ogni esame salvato in archivio.

Studente: contiene le informazioni sullo studente che sta utilizzando l'applicazione; mette a disposizione di qualsiasi classe le istanze di **Statistiche** e la lista di **Esame** (esami sostenuti) relative allo studente; offre metodi *wrapper* per aggiungere, modificare e eliminare esami al fine di mantenere aggiornate le **Statische** e provvedere alla separazione tra *frontend* e *backend*.

Statistiche: ottiene dalla configurazione le informazioni sul calcolo delle statisitche; restituisce la media ponderata degli esami sostenuti dallo studente che sta utilizzando l'applicazione; restituisce i crediti rimanenti al completamento del corso di laurea; restituisce la previsione (attuale, migliore, peggiore) sul voto di laurea dello studente; restituisce la lista della media dei voti per ogni esame nell'archivio.

Esame: classe *bean* che trasporta le informazioni su un esame; mette a disposizione di qualsiaisi classe i metodi per accedere e modificare tali informazioni; offre un metodo *wrapper* per ottenere un **Esame** da **ArchivioEsami** a partire dal codice esame.

ArchivioEsami: restituisce la lista degli esami di uno studente; restituisce la media dei voti per ogni esame salvato in archivio; permette di aggiungere, modificare e eliminare gli esami salvati in archivio; permette di ottenere un **Esame** a partire dal codice esame.

ParametriDiConfigurazione: legge il file di configurazione XML; invoca la validazione del file di configurazione XML, deserializza il contenuto XML come oggetti Java (istanza di **ParametriDiConfigurazione**, **Studente**, **CorsoDiLaurea** e **ParametriDiConnessione**); mette a disposizione di qualsiasi classe tutti i parametri di configurazione comprese le istanze di **Studente**, **CorsoDiLaurea** e **ParametriDiConnessione**.

CorsoDiLaurea: si (de)serializza in XML; mette a disposizione di qualsiasi classe i parametri di configurazione riguardanti il corso di laurea.

ParametriDiConnessione: si (de)serializza in XML; mette a disposizione di qualsiasi classe i parametri di configurazione riguardanti la connessione con **ServerLogger**.

InputCache: contiene su file binario la riga seleziona in **TabellaEsami** e gli input inseriti in **AreaModifica**; offre i metodi per prelevare e salvare tali dati; offre un metodo per prelevare l'input dell'utente come istanza di **Esame**.

EventoDiNavigazione: *(server e client)* contiene le informazioni di un evento di log; si serializza in XML; contiene i metodi per l'invio, la ricezione e la validazione degli eventi di log tramite XML Schema.

DataOraEvento: *(server e client)* rappresenta la data o l'ora di un **EventoDiNavigazione**, con relativo formato di rappresentazione.

DataOraEventoConverter: *(server e client)* converter per il corretto *marshalling* XML della classe **DataOraEvento** con **XStream**.

ServerLogger: *(server)* riceve un evento di log XML e ne invoca la validazione dell'evento di log tramite la classe **EventoDiNavigazione**; aggiunge la riga XML al file di log in modo incrementale.

Architettura del Software