# Documento di Progetto – Responsabilità delle Classi

Classe **GestioneMonetariaFinestra**

* Eredita Application
* Gestisce la componente grafica dell’applicazione in ogni sfumatura
* Definisce gli handler per i componenti grafici
* Sfrutta varie classi qui elencate per la gestione della cache, del logging, del caricamento del file di configurazione e per la gestione delle componenti grafiche

Classe **GraficoStatisticheMonetarie**

* Eredita LineChart
* Si occupa di popolare e mantenere aggiornata l’istanza del grafico a partire da una lista di GuadagnoSpesa

Classe **GuadagnoSpesa**

* Rappresenta una voce monetaria di GuadagnoSpesa, mantenendo tutti i membri privati
* Ha una serie di metodi get per ottenere i membri dell’istanza

Classe **OperazioniDatabaseGuadagniSpese**

* Contiene una serie di metodi per gestire il flusso di informazioni da e verso il database
* In particolare ha un metodo per aggiungere una voce di GuadagnoSpesa nel database e due metodi per ottenere le voci contenute in esso, di cui uno con filtro e l’altro senza

Classe **TabellaVisualeGuadagniSpese**

* Ha il compito di mantenere aggiornata la tabella di GuadagniSpese, sfruttando la classe bean GuadagnoSpesa e una lista osservabile di GuadagnoSpesa

Classe **FileGuadagniSpeseXML**

* Ha il compito di caricare e validare una lista di GuadagniSpese da un file XML, restituendola al chiamante
* Sfrutta la classe ValidatoreXML per validare il file caricato, tramite un apposito schema XSD

Classe **CacheGestioneMonetaria**

* Mantiene una cache delle principali componenti dell’interfaccia grafica, tra le quali le voci del filtro e quelle dell’inserimento voce monetaria
* E’ serializzabile
* Ha un metodo che si occupa di applicare le voci in cache sulle relative componenti dell’interfaccia grafica
* Si occupa di salvare e caricare la cache su/da un file binario nella directory principale del programma

Classe **ConfigurazioneXML**

* Contiene i parametri di configurazione dell’applicazione
* E’ serializzabile
* Ha un metodo per caricare i parametri di configurazione da un file XML, non prima di averlo validato sfruttando la classe ValidatoreXML e un apposito file XSD come schema

Classe **ValidatoreXML**

* Ha un solo metodo statico che si occupa di verificare la validità di un file XML sulla base di uno schema XSD

Classe **ElementoLog**

* Contiene gli elementi che descrivono una voce di log come membri della classe
* E’ serializzabile
* E’ usata dalle classi che inviano/ricevono voci di log

Classe **InvioDatiLog**

* Serializza le voci di log create dalla classe GestioneMonetariaFinestra e le invia al server di log servendosi dei socket di rete

Classe **LogServerXML**

* E’ la classe principale del Server di Log
* Si occupa di accettare richieste TCP da parte dell’applicazione GestioneMonetaria
* Valida il risultato XML ricevuto tramite un apposito file XSD
* Visualizza la voce di log nella console dell’applicazione e la salva in un file sempre aperto in scrittura, che non viene validato