*Descrizione delle classi*

Funzioni di base: il programma ruota intorno all’emissione di Modelli F24 (con particolare accento sul pagamento della quota IRAP) da parte di aziende, previamente registrate a sistema, e presso comuni, anch’essi previamente registrati a sistema.

Sono definite le sopracitate tre classi di oggetti:

* Impresa:

Per essere correttamente registrata a sistema ed avere la possibilità di emettere un Modello F24 presso un comune di riferimento, deve possedere i seguenti attributi:

1. Codice fiscale (assunto come univoco)
2. Denominazione ufficiale
3. Ragione Sociale
4. Codice di Divisione ATECO
5. Numero di dipendenti
6. Numero di soci
7. Numero di amministratori
8. Presenza o assenza di certificati di qualità
9. Fatturato annuale

* Comune:

Per essere correttamente registrato a sistema e poter emettere Modelli F24 (o permettere all’utente di visualizzare tutti i modelli precedentemente emessi), deve possedere i seguenti attributi:

1. Nome (assunto come univoco)
2. Regione di appartenenza
3. Numero di abitanti

Ciascuna istanza della classe ‘Comune’ contiene inoltre informazioni su:

* Imprese registrate presso il comune
* Storico dei modelli F24 emessi
* Modello F24:

Per essere correttamente creato (dunque emesso) e registrato a sistema deve possedere i seguenti attributi:

1. Impresa emittente
2. Data di emissione

Ciascuna istanza della classe ‘ModelloF24’ contiene inoltre l’informazione sull’iporto IRAP da pagare per l’azienda in questione, tenendo conto del suo fatturato annuo ed eventuali agevolazioni.

* Importo IRAP

*Guida d’uso*

Il programma si compone di 4 moduli Python:

1. ‘Classes’: il modulo contiene le tre classi ‘Impresa’, ‘Comune’ e ‘ModelloF24’.
   1. Moduli della classe ‘Impresa’:
      1. Costruttore
      2. ‘calcolaIRAP’: il metodo tiene conto del fatturato annuale dell’impresa e dell’eventuale presenza di agevolazioni (invocando il metodo ‘dirittoAgevolazione’) per restituire l’importo IRAP da pagare.
      3. ‘dirittoAgevolazione’: sulla base della Divisione ATECO e sul numero di dipendenti dell’impresa, il metodo restituisce ‘true’ se quest’ultima ha diritto a quote di agevolazione, false altrimenti.
   2. Moduli della classe ‘Comune’:
      1. Costruttore
      2. ‘registraImpresa’: il metodo, noto l’array di imprese registrate presso il comune in question, aggiunge una nuova istanza dell’oggetto Impresa (data in input) alla lista.
      3. ‘emettiModelloF24’: il metodo prende in input un’istanza di un oggetto Impresa e la data di emissione per creare un ‘ModelloF24’ (invocando un metodo di tale classe) e successivamente aggiungerlo all’array di Modelli F24 emessi dal comune. Prima di creare l’oggetto di tipo Modello, effettua un controllo che l’impresa data in input non abbia emesso anteriormente un modello nella stessa data.
   3. Moduli della classe ‘ModelloF24’:
      1. Costruttore
      2. ‘preparaF24’: prende in input il Modello F24 stesso e ne visualizza tutti gli attributi.
2. ‘Test1’ - Il modulo contiene i seguenti metodi:
   1. ‘leggiImprese(txt)’

Il metodo prende in input un file .txt dove sono elencati e separati da virgole tutti gli attributi di istanze di oggetti di tipo Impresa, e restituisce un array contentente tali oggetti.

* 1. ‘creaDizionarioPersoneCoinvolte(arrayImprese)’

Il metodo prende in input il preesistente array di Imprese già registrate a sistema e ritorna un dizionario contenente la somma di numero di dipendenti, soci ed amministratori per ciascuna Impresa nell’array.

* 1. ‘ordinaImprese(arrayImprese)’

Il metodo prende in input il preesistente array di Imprese già registrate a sistema e ritorna lo stesso, ma ordinato in base alla divisione ATECO e, a partià di divisione ATECO, per numero di amministratori decrescente.

* 1. ‘visualizzaImprese(arrayImprese)’

Il metodo prende in input il preesistente array di Imprese già registrate a sistema e ne stampa il contenuto, includendo tutti gli attributi per ciascuna istanza dell’oggetto Impresa.

* 1. ‘scriviImprese(arrayImprese)’

Il metodo prende in input un array di Imprese già registrate a sistema e crea un file .txt (allocandolo nella directory) dove ne salva il contenuto (tutti gli attributi impresa per impresa, separati da virgole).

* 1. ‘main()’

Il metodo richiama tutte le funzioni appena descritte e ne visualizza gli output nel terminale.

1. ‘Test2’ - Il modulo contiene i seguenti metodi:
   1. ‘calcoloMediaArtimetica’ e ‘calcoloMediaAritmetica2’

Entrambi i metodi prendono in input l’array di Imprese già registrate a sistema, ma danno output differenti. Rispettivamente:

* + ‘calcoloMediaArtimetica’ ritorna, per **ciascuna** impresa nell’array, la media tra il numero di soci e il numero di dipendenti.
  + ‘calcoloMediaArtimetica2’ ritorna solo due valori: per **tutte** le imprese nell’array, la media totale del numero dei soci e la media totale del numero dei dipendenti.
  1. ‘percentualeImprese(arrayImprese)’

Il metodo prende in input il preesistente array di Imprese già registrate a sistema e ritorna la percentuale di imprese che possiedono certificazioni di qualità e con fatturato compreso tra 10.000€ e 50.000€ all’anno.

* 1. ‘QualitaCapitale(arrayImprese)’

Il metodo prende in input il preesistente array di Imprese già registrate a sistema e ritorna una lista con la denominazione delle sole imprese che possiedono certificazioni di qualità e la cui ragione sociale è “Società di Capitale”.

* 1. ‘numeroATECO(arrayImprese)’

Il metodo prende in input il preesistente array di Imprese già registrate a sistema e ritorna, per ogni divisione ATECO, il numero totale di imprese.

* 1. ‘main()’

Il metodo richiama tutte le funzioni appena descritte e ne visualizza gli output nel terminale.

1. ‘Test3’ – Il modulo contiene i seguenti metodi:
   1. ‘creaImpresa(arrayImprese)’

Il metodo consente di creare a sistema una nuova impresa sulla base di input da tastiera. Prende in input il preesistente array di Imprese già registrate a sistema, per verificare che l’impresa non sia già presente ed aggiungerla qualora sia inedita, e ritorna lo stesso, aggiornato.

* 1. ‘creaComune(arrayComuni)’

Analogamente al metodo precedente, consente di creare a sistema un nuovo comune sulla base di input da tastiera. Prende in input il preesistente array di Comuni già registrati a sistema, per verificare che il comune non sia già presente ed aggiungerlo qualora sia inedito, e ritorna lo stesso, aggiornato.

* 1. ‘leggiComuni(txt)’

Analogamente al metodo ‘leggiImprese(txt)’ del Test1, prende in input un file .txt dove sono elencati e separati da virgole tutti gli attributi di istanze di oggetti di tipo Comune, e restituisce un array contentente tali oggetti.

* 1. ‘registrazioneIC(arrayImprese, arrayComuni)’

Il metodo consente di registrare un’impresa presso un comune, aggiungendo quest’ultima alla lista attributo della classe Comune.

* 1. ‘IRAP(arrayImprese)’

Il metodo calcola e stampa l’importo IRAP per l’impresa desiderata. L’input viene dato da tastiera sulla base del Codice Fiscale.

* 1. ‘emissione(arrayImprese, arrayComuni)’

Il metodo prende in input l’array di imprese e comuni precedentemente registrati a sistema, richiede venga digitato da tastiera il codice fiscale dell’impresa per cui si vuole emettere il ModelloF24 e, dopo aver verificato se l’impresa è registrata o no presso un comune, invoca il metodo ‘emettiModelloF24’ dandogli in input tale comune e la data odierna (questo metodo della classe Comune crea il modello e lo aggiunge all’array di Modelli F24 emessi dal comune).

* 1. ‘antichiModelliF24(arrayImprese,arrayComuni)’

Il metodo consente di registrare a sistema Modelli F24 emessi anteriormente alla data odierna, a patto che l’impresa sia esistente a sistema e registrata presso un comune. Gli input su quale sia l’impresa in questione e quale sia la data di emissione vengono dati da tastiera. Viene effettuato un controllo che la data sia nel passato.

* 1. ‘vecchiaData()’

Il metodo controlla che la data inserita tramite tastiera sia anteriore a quella odierna, entra in loop finché non viene inserita un’opzione valida e restituisce la data stessa.

* 1. ‘report(arrayComuni)’

Il metodo genera un report con statistiche sull’importo totale dell’IRAP riscosso dal comune in un determinato periodo di tempo.

* 1. ‘main()’

Il metodo richiama tutte le funzioni appena descritte e ne visualizza gli output nel terminale. In particolare, contiene una switch per essere più user-friendly e consentire all’utente di navigare tra le possibilità offerte dal programma.