

Programming with Python ed. 23-24
Ing. Francesco Lupia
Elaborato A

Obiettivo dell'Esame: Sviluppo di un Software per la Gestione delle Emissioni di Pagamento dell'IRAP

Introduzione

In questo progetto, si mira a sviluppare un software in Python che faciliti la gestione delle emissioni di pagamento dell'IRAP per le imprese, migliorando l'efficienza e l'accuratezza del processo.

Obiettivi:

- progettare ed implementare le classi Python “**Impresa**”, “**ModelloF24**”, e “**Comune**” con attributi e metodi appropriati.
- implementare funzionalità per calcolare l'imposta dovuta e generare moduli di pagamento.
- garantire che il software sia flessibile per adattarsi a diverse categorie di imprese.

Sviluppo:

Un elenco di possibili attributi della classe “Impresa” è il seguente:

- “Codice Fiscale”, il codice fiscale dell’azienda;
- “Denominazione”, denominazione dell’azienda;
- “Ragione Sociale”, ragione sociale dell’impresa: *Società di Capitale, Società di Persone, Imprese Individuali, Altre Forme*;
- “Divisione ATECO”, la divisione ATECO in cui opera l’azienda (es. A01, A02, ..., B05);
- “Numero di dipendenti”, il numero di dipendenti dell’azienda;
- “Numero di soci”, il numero di soci dell’azienda;
- “Numero di amministratori”, il numero di amministratori dell’azienda;
- “Data costituzione”, la data di costituzione dell’azienda;
- “Certificazioni di qualità”, vale *True* se l’impresa ha certificazioni di qualità; *False* altrimenti.
- “Fatturato”, fatturato annuo dell’azienda

A tale insieme di attributi ne possono essere aggiunti altri per la gestione delle funzionalità offerte dalla classe.

Per la suddetta classe “Impresa” definire almeno i metodi seguenti:

- *init*
il metodo costruttore della classe Impresa
- *def calcola_irap(self)*
Questo metodo calcola l’imposta IRAP che l’azienda deve pagare al comune a partire dal valore dell’attributo “Fatturato” in accordo ai valori riportati nella Tabella seguente. Il totale IRAP è ridotto del 1.5% se l’azienda opera nella “Divisione ATECO” *A03* e il numero di dipendenti è maggiore di 15.
Tale metodo dovrà restituire almeno il totale in Euro da pagare e potrà restituire altre informazioni se ritenute necessarie.

Fatturato	Coefficiente	Aliquota IRAP
10000 - 50000	1,2	4,9‰
50000 - 150000	1,5	7,6‰

> 150000	1,7	8.1‰
----------	-----	------

Per esempio, supponendo che l'azienda opera nella "Divisione ATECO" A03 e abbia un fatturato di 60000 Euro con 16 dipendenti:

1. Applicazione del coefficiente: $60000 * 1,5 = 90.000$ Euro;
2. Calcolo totale IRAP: $90.000 * 7,6‰ = 684,00$ Euro.
3. Applicazione della Riduzione: $684,00 \text{ Euro} - 1.5\% = \underline{673,74 \text{ Euro}}$

- *def dirittoAgevolazione(self)*. Questo metodo ritorna *True* se l'azienda è idonea a ricevere la riduzione sul totale IRAP da versare; *False* altrimenti. Ha diritto a ricevere la riduzione l'azienda che opera nella "Divisione ATECO" A03 con numero di dipendenti è maggiore di 15.

Oltre ai due suddetti metodi, definire tutti gli ulteriori metodi che si riterranno utili all'implementazione delle funzionalità richieste dalla Classe "**Impresa**".

Un elenco di possibili attributi della classe "**ModelloF24**" è il seguente:

- "impresa" (oggetto di tipo Impresa): l'impresa a cui il modello F24 si riferisce
- "importo_irap", l'importo dell'IRAP calcolato per l'impresa

Per la suddetta Classe "ModelloF24" definire almeno i metodi seguenti:

- *init*
il metodo costruttore della classe Impresa
- *def prepara_f24(self)*
il metodo che prepara il modello F24 per l'impresa, utilizzando l'importo dell'IRAP calcolato

Un elenco di possibili attributi della classe "**Comune**" è il seguente:

- "imprese_registrate" (lista di oggetti Impresa), la lista delle imprese registrate presso il comune
- "modelli_f24_emessi" (lista di oggetti ModelloF24), la lista dei modelli F24 emessi dal comune per le imprese registrate
- *init*
il metodo costruttore della classe Comune
- *def registra_impresa(self, impresa)*
il metodo che registra un'impresa presso il comune
- *def emetti_modello_f24(self, impresa)*
il metodo che emette un modello F24 per l'impresa specificata, utilizzando la classe ModelloF24 e aggiungendolo alla lista modelli_f24_emessi

Implementazione: Codificare le classi e i metodi seguendo le specifiche. Utilizzare scenari di test per verificare la correttezza.

Esempio di utilizzo del programma:

Implementare un primo programma di test che:

- Legga dal file di testo "imprese.txt" (da creare opportunamente con un editor di testo e contenente almeno i dati di 8 imprese) l'insieme delle imprese e li memorizzi, come oggetti di tipo "Impresa", all'interno di una lista;
- A partire dalla lista delle imprese, crei un dizionario contenente per ogni "Codice Fiscale" il totale di persone coinvolte nell'impresa, cioè la somma di "Numero di dipendenti", "Numero di soci", e "Numero di amministratori";

- Ordini la lista delle imprese per “Divisione ATECO” (seguendo l’ordine alfabetico) e a parità di “Divisione ATECO” per “Numero di amministratori” (in ordine decrescente);
- Visualizzi i dati di tutte le imprese presenti nella lista;
- Salvi sul file “imprese_ordinate.txt” il contenuto finale della lista.

Implementare un secondo programma di test che:

a partire dalla lista delle imprese ottenuta leggendo il file “imprese_ordinate.txt” generato in precedenza ed utilizzando opportune funzioni ausiliarie, calcoli:

- La media aritmetica del “Numero di dipendenti” e del “Numero di soci” delle imprese;
- La percentuale di imprese con “Certificazioni di qualità” o “Fatturato” compreso tra 10000 e 50000 Euro (estremi compresi);
- Le aziende con certificazioni di qualità (“Certificazioni di qualità” == True) e “Ragione Sociale” = *Società di Capitale*; utilizzare le funzionalità offerte dalla libreria Pandas;
- Per ogni “Divisione ATECO” il numero di aziende.

Implementare un terzo programma di test che:

- Crea uno o più comuni
- Effettua la registrazione di un’impresa presso il comune
- Calcoli l’import IRAP per l’impresa
- Emissione del modello F24 per l’impresa da parte del comune
- Stampa dei modelli F24 emessi dal comune
- Generazione di un report con statistiche sull’importo totale dell’IRAP riscosso dal comune in un determinato periodo di tempo.

Criteri di Valutazione:

- Adesione ai principi di programmazione orientata agli oggetti (OOP).
- Correttezza e completezza delle funzionalità implementate.
- Creatività e originalità nell’approccio alla soluzione del problema.
- Qualità generale della presentazione del lavoro, inclusa la chiarezza della scrittura e l’organizzazione del programma.

Estensioni Facoltative:

- Integrazione con database per la gestione dei dati.
- Sviluppo di un’interfaccia grafica per l’interazione con l’utente eventualmente nella forma di web application.

Consegna:

Il progetto deve essere consegnato sotto forma di uno o più moduli python (o Jupyter notebook) che includa codice eseguibile, commenti dettagliati delle funzioni implementate e una guida all’uso del programma. Nel caso in cui il progetto sia composto da più file, si richiede di creare un archivio esclusivamente in **formato zip**. Avvisare dell’avvenuta consegna del progetto tramite Teams (le email potrebbero finire nello spam a causa degli allegati).

Email per consegna progetto: francesco.lupia@unical.it

La deadline per la consegna è **15/04/2024**.

Buon lavoro!