Progetto di Programmazione Dichiarativa (a.a. 2024/25)

Realizzato da Colli Simone e Merenda Saverio Mattia

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 main Namespace Reference

Functions

- def estrazione_dati_per_ssd (path=path_coperture)
- def init ()
- def init_corsi (filename)
- def init_corsi_matricole (filepathCorsi, filepathProf)
- def init_matricole (filename)
- def main ()
- def main_old ()
- def process_ssd (dataset_loader, ssd, failed_ssds)
- def run_tests ()
- def scrittura_fatti (filters=None, file_name=None, path=path_coperture)

  FUNZIONI PRINCIPALI ##.
- def test_estrazione_settori ()
- def test_filtra_per_colonne ()
- def test_filtra_per_valori ()
- def test_parser ()

  TEST ##.
- def unisci_file_per_settore (path)

Variables

- string path_coperture = "dataset/coperture.xlsx"

  VARIABILI GLOBALI ################################.
- string path_docenti = "dataset/docenti.xlsx"
- string path_docenti_a_contratto = "dataset/docenti_a_contratto.xlsx"

4.1.1 Function Documentation

4.1.1.1 estrazione_dati_per_ssd()

```
def main.estrazione_dati_per_ssd (
              path = path_coperture )
```

Estrae i dati per ogni SSD, applicando il filtro e salvando i file CSV nella cartella 'dataset/ssd/'.

Questa funzione esegue le seguenti operazioni:
1. Carica i dati da un file Excel e seleziona la colonna 'SSD', rimuovendo i duplicati.
2. Crea una barra di progresso per monitorare l'elaborazione degli SSD.
3. Utilizza un `ThreadPoolExecutor` per elaborare i dati in parallelo, migliorando le performance.
4. Crea la cartella di output 'dataset/ssd/' se non esiste già.
5. Per ogni SSD, chiama la funzione `process_ssd` per applicare il filtro e salvare il file CSV corrispondente
6. Gestisce gli errori durante l'elaborazione e registra gli SSD che non sono stati salvati.

:param dataset_loader: Istanza di `DatasetLoader` per accedere ai metodi di filtraggio e salvataggio.
:param failed_ssds: Set per raccogliere gli SSD che causano errori durante l'elaborazione.

4.1.1.2 init()

```
def main.init ( )
```

4.1.1.3 init_corsi()

```
def main.init_corsi (
              filename )
```

4.1.1.4 init_corsi_matricole()

```
def main.init_corsi_matricole (
              filepathCorsi,
              filepathProf )
```

4.1.1.5 init_matricole()

```
def main.init_matricole (
              filename )
```

4.1.1.6 main()

```
def main.main ( )
```

4.1.1.7 main_old()

```
def main.main_old ( )
```

4.1.1.8 process_ssd()

```
def main.process_ssd (
              dataset_loader,
              ssd,
              failed_ssds )
```

```
Elabora un singolo valore SSD, applicando il filtro e salvando il file.
:param dataset_loader: Istanza di DatasetLoader.
:param ssd: Valore dell'SSD da elaborare.
:param failed_ssds: Set per raccogliere gli SSD che causano errori.
```

4.1.1.9 run_tests()

```
def main.run_tests ( )
```

4.1.1.10 scrittura_fatti()

```
def main.scrittura_fatti (
              filters = None,
              file_name = None,
              path = path_coperture )
```

FUNZIONI PRINCIPALI ##.

```
Filtra i dati da un dataset Excel basandosi sui criteri forniti e scrive i fatti in un file ASP.

:param filters: Dizionario contenente i criteri di filtraggio.
                Esempio:
                {
                    'Cod. Tipo Corso': ['LM'],
                    'SSD': ['INF/01']
                }
                Se non specificato, vengono utilizzati i valori predefiniti:
                {'Cod. Tipo Corso': ['LM'], 'SSD': ['INF/01']}.
:param file_name: Nome del file di output (senza estensione) dove verranno salvati i fatti generati.
                  Se non specificato, il file verrà salvato con il nome predefinito "test".

Funzionamento:
1. Inizializza un'istanza di `DatasetLoader` per caricare i dati dal file `dataset/coperture.xlsx`.
2. Filtra i dati in base ai criteri specificati in `filters`.
3. Inizializza un'istanza di `DatasetManager`.
4. Utilizza il metodo `write_atoms` del `DatasetManager` per scrivere i fatti filtrati nel file ASP specificat

Esempio:
--------
Se il dataset contiene corsi con SSD 'INF/01' e tipo corso 'LM', e il filtro fornito è:
filters = {
    'Cod. Tipo Corso': ['LM'],
    'SSD': ['INF/01']
}
e `file_name = "informatica"`, i dati filtrati verranno salvati nel file `informatica.lp`.

:raises FileNotFoundError: Se il file specificato nel DatasetLoader non esiste.
:raises Exception: Per altri errori durante il filtraggio o la scrittura dei fatti.
```

4.1.1.11 test_estrazione_settori()

```
def main.test_estrazione_settori ( )
```

4.1.1.12 test_filtra_per_colonne()

```
def main.test_filtra_per_colonne ( )
```

4.1.1.13 test_filtra_per_valori()

```
def main.test_filtra_per_valori ( )
```

4.1.1.14 test_parser()

```
def main.test_parser ( )
```

TEST ##.

4.1.1.15 unisci_file_per_settore()

```
def main.unisci_file_per_settore (
              path )
```

```
Unisce i file CSV che hanno lo stesso settore in base al prefisso del nome file.
Salva il risultato in un unico file per settore senza duplicare l'intestazione.
Sposta i file originali con '_' nel nome nella cartella 'caratterizzanti'.

:param path: Percorso alla cartella contenente i file CSV.
```

4.1.2 Variable Documentation

4.1.2.1 path_coperture

```
string path_coperture = "dataset/coperture.xlsx"
```

VARIABILI GLOBALI ##################################.

4.1.2.2 path_docenti

```
string path_docenti = "dataset/docenti.xlsx"
```

4.1.2.3 path_docenti_a_contratto

```
string path_docenti_a_contratto = "dataset/docenti_a_contratto.xlsx"
```

4.2 modules Namespace Reference

Namespaces

- course_parser
- dataset_loader
- dataset_manager
- department_parser

4.3 modules.course_parser Namespace Reference

Classes

- class CourseParser

4.4 modules.dataset_loader Namespace Reference

Classes

- class DatasetLoader

4.5 modules.dataset_manager Namespace Reference

Classes

- class DatasetManager

4.6 modules.department_parser Namespace Reference

Classes

- class DepartmentParser

4.7 sanitize Namespace Reference

Functions

- def aggiorna_cod_tipo_corso (row)
- def compute_extra_data ()
- def compute_remained ()
- def sanitize ()
- def sanitize_codici_corso (df)
- def sanitize_coperture ()
- def sanitize_docenti ()

Variables

- string path_coperture = "dataset/coperture.xlsx"
- string path_coperture_contratti = "dataset/docenti_a_contratto.xlsx"
- string path_coperture_rimaste = "dataset/insegnamenti_senza_docente.xlsx"
- string path_docenti = "dataset/docenti.xlsx"
- string path_ns_coperture = "dataset/originali/coperture.xlsx"
- string path_ns_docenti = "dataset/originali/docenti.xlsx"

4.7.1 Function Documentation

4.7.1.1 aggiorna_cod_tipo_corso()

```
def sanitize.aggiorna_cod_tipo_corso (
              row )
```

4.7.1.2 compute_extra_data()

```
def sanitize.compute_extra_data ( )
```

4.7.1.3 compute_remained()

```
def sanitize.compute_remained ( )
```

4.7.1.4 sanitize()

```
def sanitize.sanitize ( )
```

4.7.1.5 sanitize_codici_corso()

```
def sanitize.sanitize_codici_corso (
              df )
```

4.7.1.6 sanitize_coperture()

```
def sanitize.sanitize_coperture ( )
```

4.7.1.7 sanitize_docenti()

```
def sanitize.sanitize_docenti ( )
```

4.7.2 Variable Documentation

4.7.2.1 path_coperture

```
string path_coperture = "dataset/coperture.xlsx"
```

4.7.2.2 path_coperture_contratti

```
string path_coperture_contratti = "dataset/docenti_a_contratto.xlsx"
```

4.7.2.3 path_coperture_rimaste

```
string path_coperture_rimaste = "dataset/insegnamenti_senza_docente.xlsx"
```

4.7.2.4 path_docenti

```
string path_docenti = "dataset/docenti.xlsx"
```

4.7.2.5 path_ns_coperture

```
string path_ns_coperture = "dataset/originali/coperture.xlsx"
```

4.7.2.6 path_ns_docenti

```
string path_ns_docenti = "dataset/originali/docenti.xlsx"
```

Chapter 5

Class Documentation

5.1 CourseParser Class Reference

Public Member Functions

- def __init__ (self)
- def add_courses (self, courses)
- def parse (self)

Public Attributes

- parser

5.1.1 Constructor & Destructor Documentation

5.1.1.1 __init__()

```
def __init__ (
               self )
```

Inizializza il parser per i corsi.

5.1.2 Member Function Documentation

5.1.2.1 add_courses()

```
def add_courses (
              self,
              courses )
```

```
Aggiunge argomenti dinamici per ciascun dipartimento.
:param departments: Lista dei corsi.
```

5.1.2.2 parse()

```
def parse (
              self )
```

```
Effettua il parsing degli argomenti.
:return: Gli argomenti parsati.
```

5.1.3 Member Data Documentation

5.1.3.1 parser

```
parser
```

The documentation for this class was generated from the following file:

- src/modules/course_parser.py

5.2 DatasetLoader Class Reference

Public Member Functions

- def __init__ (self, input_file_path="dataset/coperture.xlsx")
- def df_to_dict (self, df, id_type="col")
- def filter_by_values (self, filters, dataset=None, only_prefix=False)
- def get_values (self, dataset=None, columns=None)
- def save_to_file (self, df, output_file_path, file_format="csv")

Public Attributes

- input_file_path

5.2.1 Detailed Description

```
Classe per la gestione del caricamento, filtraggio e salvataggio dei dati da file Excel o CSV.
```

5.2.2 Constructor & Destructor Documentation

5.2.2.1 __init__()

```
def __init__ (
              self,
              input_file_path = "dataset/coperture.xlsx" )
```

```
Inizializza la classe DatasetLoader con il percorso al dataset.
:param input_file_path: Path relativo al file Excel da leggere (default: "dataset/coperture.xlsx").
```

5.2.3 Member Function Documentation

5.2.3.1 df_to_dict()

```
def df_to_dict (
              self,
              df,
              id_type = "col" )
```

```
Converte un DataFrame Pandas in un dizionario strutturato.

Il dizionario risultante contiene:
- "rows": lista degli identificativi delle righe (index del DataFrame).
- "cols": lista degli identificativi delle colonne (header del DataFrame).
- Chiavi dinamiche: il contenuto delle colonne (default) o delle righe.

:param df: Un DataFrame Pandas da convertire.
:type df: pandas.DataFrame
:param id_type: Tipo di identificatore per il contenuto. "col" per colonne (default),
                "row" per righe.
:type id_type: str, optional
:return: Un dizionario strutturato contenente informazioni su righe, colonne
         e i dati associati.
:rtype: dict

:example:
input:
    df =
        A   B   C
    0   1   2   3
    1   4   5   6

output (id_type="col"):
{
    "rows": [0, 1],
    "cols": ["A", "B", "C"],
```

```
    "A": [1, 4],
    "B": [2, 5],
    "C": [3, 6]
}

output (id_type="row"):
{
    "rows": [0, 1],
    "cols": ["A", "B", "C"],
    0: [1, 2, 3],
    1: [4, 5, 6]
}
```

5.2.3.2 filter_by_values()

```
def filter_by_values (
              self,
              filters,
              dataset = None,
              only_prefix = False )
```

```
Filtra i dati in base a più valori specifici in diverse colonne, convertendo tutti i valori in stringhe.
:param filters: Dizionario in cui le chiavi sono i nomi delle colonne e i valori sono liste di valori da filtr
                Esempio: {'Cod. Tipo Corso': ['LM'], 'SSD': ['INF/01', 'MAT/05']}.
:param dataset: DataFrame esistente da utilizzare (default: None). Se None, viene caricato il file Excel.
:return: DataFrame contenente solo le righe che soddisfano i criteri di filtraggio.
:raises ImportError: Se il pacchetto openpyxl non è installato.
:raises KeyError: Se una delle colonne specificate nei filtri non esiste nel dataset.
:raises Exception: Per altri errori durante il filtraggio.
```

5.2.3.3 get_values()

```
def get_values (
              self,
              dataset = None,
              columns = None )
```

```
Carica un file Excel o utilizza un dataset esistente, filtrando opzionalmente per colonne specifiche.
:param dataset: DataFrame esistente da utilizzare (default: None). Se None, viene caricato il file Excel.
:param columns: Lista di colonne da mantenere (esempio: ['Matricola', 'Cognome']).
:return: DataFrame contenente tutte le righe, filtrato per le colonne specificate (se fornite).
:raises ImportError: Se il pacchetto openpyxl non è installato.
:raises KeyError: Se una o più colonne specificate non esistono nel dataset.
```

5.2.3.4 save_to_file()

```
def save_to_file (
              self,
              df,
              output_file_path,
              file_format = "csv" )
```

```
Salva il DataFrame su un file specificato.
:param df: Il DataFrame da salvare.
:param output_file_path: Percorso del file di output.
:param file_format: Formato del file, può essere 'csv' o 'excel'.
:raises ValueError: Se il formato specificato non è 'csv' o 'excel'.
:raises Exception: Per altri errori durante il salvataggio del file.
```

5.2.4 Member Data Documentation

5.2.4.1 input_file_path

```
input_file_path
```

The documentation for this class was generated from the following file:

- src/modules/dataset_loader.py

5.3 DatasetManager Class Reference

Public Member Functions

- def __init__ (self, dataset_path="dataset/")
- def get_courses (self)
- def get_departments (self)
- def get_professors (self)
- def get_sectors (self, SSD=None)
- def load_data (self, filename)
- def scrivi_coperture (self, df, filename)
- def scrivi_docenti (self, df, filename)
- def scrivi_docenti_a_contratto (self, df, filename)

Public Attributes

- dataset_path

5.3.1 Detailed Description

```
Classe per la gestione di file di dataset e la generazione di file ASP strutturati.
```

5.3.2 Constructor & Destructor Documentation

5.3.2.1 __init__()

```
def __init__ (
              self,
              dataset_path = "dataset/" )
```

Inizializza la classe DatasetManager con il percorso della cartella contenente i file di dataset.
:param dataset_path: Path alla cartella contenente i file di dataset (default: "dataset/").

5.3.3 Member Function Documentation

5.3.3.1 get_courses()

```
def get_courses (
              self )
```

Ottiene und dizionario {"codice_corso": "corso"} basandosi sui file presenti nella cartella del dataset.
:return: Un dizionario di corsi.
:raises FileNotFoundError: Se la cartella del dataset non esiste.

5.3.3.2 get_departments()

```
def get_departments (
              self )
```

Ottiene la lista dei dipartimenti basandosi sui file presenti nella cartella del dataset.
:return: Una lista di dipartimenti (nomi dei file senza estensione).
:raises FileNotFoundError: Se la cartella del dataset non esiste.

5.3.3.3 get_professors()

```
def get_professors (
              self )
```

Ottiene und dizionario {"codice_corso": {"matricola"}} basandosi sui file presenti nella cartella del dataset.
:return: Un dizionario di corsi: set_matricole.
:raises FileNotFoundError: Se la cartella del dataset non esiste.

5.3.3.4 get_sectors()

```
def get_sectors (
              self,
              SSD = None )
```

```
Ottiene la lista dei settori partendo da una lista di SSD.

Ogni SSD è rappresentato nel formato "SETTORE/CODICE". Questa funzione estrae
e restituisce solo la parte relativa al settore (prima del separatore '/').

:param SSD: Una lista di stringhe rappresentanti i SSD. Ad esempio, ["INF/01", "MAT/03"].
:type SSD: list[str], optional
:return: Un set di settori se la lista passata come parametro è non vuota.
        Ritorna un set vuoto se il parametro è None o vuoto.
:rtype: set[str]

:example:
input: ["INF/01", "MAT/03", "FIS/07"]
output: {"INF", "MAT", "FIS"}
```

5.3.3.5 load_data()

```
def load_data (
              self,
              filename )
```

```
Legge i dati da un file di testo e li stampa.
:param filename: Nome del file (senza estensione) da cui leggere i dati.
:raises FileNotFoundError: Se il file specificato non esiste.
:raises Exception: Per altri errori durante la lettura del file.
```

5.3.3.6 scrivi_coperture()

```
def scrivi_coperture (
              self,
              df,
              filename )
```

```
Scrive i dati del DataFrame in un file ASP (.lp), organizzandoli per sezioni con eliminazione di duplicati.
:param df: DataFrame contenente i dati da salvare.
:param filename: Nome del file di output (senza estensione).
:raises Exception: Per errori durante la scrittura del file.
```

5.3.3.7 scrivi_docenti()

```
def scrivi_docenti (
              self,
              df,
              filename )
```

```
Scrive i dati del DataFrame in un file ASP (.lp), organizzandoli per sezioni con eliminazione di duplicati.
:param df: DataFrame contenente i dati da salvare.
:param filename: Nome del file di output (senza estensione).
:raises Exception: Per errori durante la scrittura del file.
```

5.3.3.8 scrivi_docenti_a_contratto()

```
def scrivi_docenti_a_contratto (
              self,
              df,
              filename )
```

```
Scrive i dati del DataFrame in un file ASP (.lp), organizzandoli per sezioni con eliminazione di duplicati.
:param df: DataFrame contenente i dati da salvare.
:param filename: Nome del file di output (senza estensione).
:raises Exception: Per errori durante la scrittura del file.
```

5.3.4 Member Data Documentation

5.3.4.1 dataset_path

```
dataset_path
```

The documentation for this class was generated from the following file:

- src/modules/dataset_manager.py

5.4 DepartmentParser Class Reference

Public Member Functions

- def __init__ (self)
- def add_departments (self, departments)
- def parse (self)

Public Attributes

- parser

5.4.1 Constructor & Destructor Documentation

5.4.1.1 __init__()

```
def __init__ (
              self )
```

```
Inizializza il parser per i dipartimenti.
```

5.4.2 Member Function Documentation

5.4.2.1 add_departments()

```
def add_departments (
              self,
              departments )
```

```
Aggiunge argomenti dinamici per ciascun dipartimento.
:param departments: Lista dei dipartimenti.
```

5.4.2.2 parse()

```
def parse (
              self )
```

```
Effettua il parsing degli argomenti.
:return: Gli argomenti parsati.
```

5.4.3 Member Data Documentation

5.4.3.1 parser

```
parser
```

The documentation for this class was generated from the following file:

- src/modules/department_parser.py

Chapter 6

File Documentation

6.1 src/main.py File Reference

Namespaces

- main

Functions

- def estrazione_dati_per_ssd (path=path_coperture)
- def init ()
- def init_corsi (filename)
- def init_corsi_matricole (filepathCorsi, filepathProf)
- def init_matricole (filename)
- def main ()
- def main_old ()
- def process_ssd (dataset_loader, ssd, failed_ssds)
- def run_tests ()
- def scrittura_fatti (filters=None, file_name=None, path=path_coperture)

  FUNZIONI PRINCIPALI ##.
- def test_estrazione_settori ()
- def test_filtra_per_colonne ()
- def test_filtra_per_valori ()
- def test_parser ()

  TEST ##.
- def unisci_file_per_settore (path)

Variables

- string path_coperture = "dataset/coperture.xlsx"

  VARIABILI GLOBALI ################################.
- string path_docenti = "dataset/docenti.xlsx"
- string path_docenti_a_contratto = "dataset/docenti_a_contratto.xlsx"

6.2 src/modules/__init__.py File Reference

Namespaces

- modules

6.3 src/modules/course_parser.py File Reference

Classes

- class CourseParser

Namespaces

- modules.course_parser

6.4 src/modules/dataset_loader.py File Reference

Classes

- class DatasetLoader

Namespaces

- modules.dataset_loader

6.5 src/modules/dataset_manager.py File Reference

Classes

- class DatasetManager

Namespaces

- modules.dataset_manager

6.6 src/modules/department_parser.py File Reference

Classes

- class DepartmentParser

Namespaces

- modules.department_parser

6.7 src/sanitize.py File Reference

Namespaces

- sanitize

Functions

- def aggiorna_cod_tipo_corso (row)
- def compute_extra_data ()
- def compute_remained ()
- def sanitize ()
- def sanitize_codici_corso (df)
- def sanitize_coperture ()
- def sanitize_docenti ()

Variables

- string path_coperture = "dataset/coperture.xlsx"
- string path_coperture_contratti = "dataset/docenti_a_contratto.xlsx"
- string path_coperture_rimaste = "dataset/insegnamenti_senza_docente.xlsx"
- string path_docenti = "dataset/docenti.xlsx"
- string path_ns_coperture = "dataset/originali/coperture.xlsx"
- string path_ns_docenti = "dataset/originali/docenti.xlsx"

Index