

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

Realizzato da Colli Simone e Merenda Saverio Mattia

Generated by Doxygen 1.9.8

1 Namespace Index	1
1.1 Package List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 main Namespace Reference	7
4.1.1 Function Documentation	8
4.1.1.1 estrazione_dati_per_ssd()	8
4.1.1.2 init()	8
4.1.1.3 init_corsi()	8
4.1.1.4 init_corsi_matricole()	8
4.1.1.5 init_matricole()	8
4.1.1.6 main()	8
4.1.1.7 main_old()	8
4.1.1.8 process_ssd()	9
4.1.1.9 run_tests()	9
4.1.1.10 scrittura_fatti()	9
4.1.1.11 test_estrusione_settori()	10
4.1.1.12 test_filtra_per_colonne()	10
4.1.1.13 test_filtra_per_valori()	10
4.1.1.14 test_parser()	10
4.1.1.15 unisci_file_per_settore()	10
4.1.2 Variable Documentation	10
4.1.2.1 path_coperture	10
4.1.2.2 path_docenti	10
4.1.2.3 path_docenti_a_contratto	10
4.1.2.4 paths_immatricolati	11
4.2 modules Namespace Reference	11
4.3 modules.course_parser Namespace Reference	11
4.4 modules.dataset_loader Namespace Reference	11
4.5 modules.dataset_manager Namespace Reference	11
4.6 modules.department_parser Namespace Reference	11
4.7 sanitize Namespace Reference	12
4.7.1 Function Documentation	12
4.7.1.1 aggiorna_cod_tipo_corso()	12
4.7.1.2 compute_extra_data()	12
4.7.1.3 compute_remained()	12
4.7.1.4 sanitize()	12

4.7.1.5 <code>sanitize_codici_corso()</code>	12
4.7.1.6 <code>sanitize_coperture()</code>	13
4.7.1.7 <code>sanitize_docenti()</code>	13
4.7.2 Variable Documentation	13
4.7.2.1 <code>path_coperture</code>	13
4.7.2.2 <code>path_coperture_contratti</code>	13
4.7.2.3 <code>path_coperture_rimaste</code>	13
4.7.2.4 <code>path_docenti</code>	13
4.7.2.5 <code>path_ns_coperture</code>	13
4.7.2.6 <code>path_ns_docenti</code>	13
5 Class Documentation	15
5.1 CourseParser Class Reference	15
5.1.1 Constructor & Destructor Documentation	15
5.1.1.1 <code>__init__()</code>	15
5.1.2 Member Function Documentation	15
5.1.2.1 <code>add_courses()</code>	15
5.1.2.2 <code>parse()</code>	16
5.1.3 Member Data Documentation	16
5.1.3.1 <code>parser</code>	16
5.2 DatasetLoader Class Reference	16
5.2.1 Detailed Description	16
5.2.2 Constructor & Destructor Documentation	16
5.2.2.1 <code>__init__()</code>	16
5.2.3 Member Function Documentation	17
5.2.3.1 <code>df_to_dict()</code>	17
5.2.3.2 <code>filter_by_values()</code>	17
5.2.3.3 <code>get_values()</code>	18
5.2.3.4 <code>save_to_file()</code>	18
5.2.4 Member Data Documentation	18
5.2.4.1 <code>input_file_path</code>	18
5.3 DatasetManager Class Reference	18
5.3.1 Detailed Description	19
5.3.2 Constructor & Destructor Documentation	19
5.3.2.1 <code>__init__()</code>	19
5.3.3 Member Function Documentation	19
5.3.3.1 <code>get_courses()</code>	19
5.3.3.2 <code>get_departments()</code>	19
5.3.3.3 <code>get_professors()</code>	19
5.3.3.4 <code>get_sectors()</code>	20
5.3.3.5 <code>load_data()</code>	20
5.3.3.6 <code>scrivi_coperture()</code>	20

5.3.3.7 scrivi_docenti()	20
5.3.3.8 scrivi_docenti_a_contratto()	21
5.3.3.9 scrivi_ministeriali()	21
5.3.4 Member Data Documentation	21
5.3.4.1 dataset_path	21
5.4 DepartmentParser Class Reference	21
5.4.1 Constructor & Destructor Documentation	22
5.4.1.1 __init__()	22
5.4.2 Member Function Documentation	22
5.4.2.1 add_departments()	22
5.4.2.2 parse()	22
5.4.3 Member Data Documentation	22
5.4.3.1 parser	22
6 File Documentation	23
6.1 src/main.py File Reference	23
6.2 src/modules/__init__.py File Reference	24
6.3 src/modules/course_parser.py File Reference	24
6.4 src/modules/dataset_loader.py File Reference	24
6.5 src/modules/dataset_manager.py File Reference	24
6.6 src/modules/department_parser.py File Reference	24
6.7 src/sanitize.py File Reference	25
Index	27

Chapter 1

Namespace Index

1.1 Package List

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

main	7
modules	11
modules.course_parser	11
modules.dataset_loader	11
modules.dataset_manager	11
modules.department_parser	11
sanitize	12

Chapter 2

Class Index

2.1 Class List

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

CourseParser	15
DatasetLoader	16
DatasetManager	18
DepartmentParser	21

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/ main.py	23
src/ sanitize.py	25
src/modules/ __init__.py	24
src/modules/ course_parser.py	24
src/modules/ dataset_loader.py	24
src/modules/ dataset_manager.py	24
src/modules/ department_parser.py	24

Chapter 4

Namespace Documentation

4.1 main Namespace Reference

Functions

- [estrazione_dati_per_ssd](#) (path=[path_coperture](#))
 - [init](#) ()
 - [init_corsi](#) (filename)
 - [init_corsi_matricole](#) (filepathCorsi, filepathProf)
 - [init_matricole](#) (filename)
 - [main](#) ()
 - [main_old](#) ()
 - [process_ssd](#) (dataset_loader, ssd, failed_ssds)
 - [run_tests](#) ()
 - [scrittura_fatti](#) (filters=None, file_name=None, path=[path_coperture](#))
- FUNZIONI PRINCIPALI #####.*
- [test_estrazione_settori](#) ()
 - [test_filtra_per_colonne](#) ()
 - [test_filtra_per_valori](#) ()
 - [test_parser](#) ()
- TEST #####.*
- [unisci_file_per_settore](#) (path)

Variables

- str [path_coperture](#) = "dataset/coperture.xlsx"
- VARIABILI GLOBALI #####.*
- str [path_docenti](#) = "dataset/docenti.xlsx"
 - str [path_docenti_a_contratto](#) = "dataset/docenti_a_contratto.xlsx"
 - list [paths_immatricolati](#) = ["dataset/immatricolati/LT.xlsx", "dataset/immatricolati/LM.xlsx", "dataset/immatricolati/CU.xlsx"]

4.1.1 Function Documentation

4.1.1.1 estrazione_dati_per_ssd()

```
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()

```
init ( )
```

4.1.1.3 init_corsi()

```
init_corsi (
    filename )
```

4.1.1.4 init_corsi_matricole()

```
init_corsi_matricole (
    filepathCorsi,
    filepathProf )
```

4.1.1.5 init_matricole()

```
init_matricole (
    filename )
```

4.1.1.6 main()

```
main ( )
```

4.1.1.7 main_old()

```
main_old ( )
```

4.1.1.8 process_ssd()

```
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()

```
run_tests ( )
```

4.1.1.10 scrittura_fatti()

```
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 specificato.

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()

```
test_estrazione_settori ( )
```

4.1.1.12 test_filtra_per_colonne()

```
test_filtra_per_colonne ( )
```

4.1.1.13 test_filtra_per_valori()

```
test_filtra_per_valori ( )
```

4.1.1.14 test_parser()

```
test_parser ( )
```

TEST #####.

4.1.1.15 unisci_file_per_settore()

```
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

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

VARIABILI GLOBALI #####.

4.1.2.2 path_docenti

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

4.1.2.3 path_docenti_a_contratto

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


4.1.2.4 paths_immatricolati

```
list paths_immatricolati = ["dataset/immatricolati/LT.xlsx", "dataset/immatricolati/LM.xlsx",  
"dataset/immatricolati/CU.xlsx"]
```

4.2 modules Namespace Reference

Namespaces

- namespace [course_parser](#)
- namespace [dataset_loader](#)
- namespace [dataset_manager](#)
- namespace [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

- [aggiorna_cod_tipo_corso](#) (row)
- [compute_extra_data](#) ()
- [compute_remained](#) ()
- [sanitize](#) ()
- [sanitize_codici_corso](#) (df)
- [sanitize_coperture](#) ()
- [sanitize_docenti](#) ()

Variables

- str [path_coperture](#) = "dataset/coperture.xlsx"
- str [path_coperture_contratti](#) = "dataset/docenti_a_contratto.xlsx"
- str [path_coperture_rimaste](#) = "dataset/insegnamenti_senza_docente.xlsx"
- str [path_docenti](#) = "dataset/docenti.xlsx"
- str [path_ns_coperture](#) = "dataset/originali/coperture.xlsx"
- str [path_ns_docenti](#) = "dataset/originali/docenti.xlsx"

4.7.1 Function Documentation

4.7.1.1 [aggiorna_cod_tipo_corso\(\)](#)

```
aggiorna_cod_tipo_corso (  
    row )
```

4.7.1.2 [compute_extra_data\(\)](#)

```
compute_extra_data ( )
```

4.7.1.3 [compute_remained\(\)](#)

```
compute_remained ( )
```

4.7.1.4 [sanitize\(\)](#)

```
sanitize ( )
```

4.7.1.5 [sanitize_codici_corso\(\)](#)

```
sanitize_codici_corso (  
    df )
```

4.7.1.6 `sanitize_coperture()`

```
sanitize_coperture ( )
```

4.7.1.7 `sanitize_docenti()`

```
sanitize_docenti ( )
```

4.7.2 Variable Documentation

4.7.2.1 `path_coperture`

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

4.7.2.2 `path_coperture_contratti`

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

4.7.2.3 `path_coperture_rimaste`

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

4.7.2.4 `path_docenti`

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

4.7.2.5 `path_ns_coperture`

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

4.7.2.6 `path_ns_docenti`

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


Chapter 5

Class Documentation

5.1 CourseParser Class Reference

Public Member Functions

- [__init__](#) (self)
- [add_courses](#) (self, courses)
- [parse](#) (self)

Public Attributes

- [parser](#)

5.1.1 Constructor & Destructor Documentation

5.1.1.1 __init__()

```
__init__ (
    self )
```

Inizializza il parser per i corsi.

5.1.2 Member Function Documentation

5.1.2.1 add_courses()

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

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

5.1.2.2 parse()

```
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

- [__init__](#) (self, [input_file_path](#)="dataset/coperture.xlsx")
- [df_to_dict](#) (self, df, id_type="col")
- [filter_by_values](#) (self, filters, dataset=None, only_prefix=False)
- [get_values](#) (self, dataset=None, columns=None)
- [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__()

```
__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()

```
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()

```
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 filtrare.
                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()

```
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()

```
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

- [__init__](#) (self, [dataset_path](#)="dataset/")
- [get_courses](#) (self)
- [get_departments](#) (self)
- [get_professors](#) (self)
- [get_sectors](#) (self, [SSD](#)=None)
- [load_data](#) (self, [filename](#))
- [scrivi_copertura](#) (self, [df](#), [filename](#))
- [scrivi_docenti](#) (self, [df](#), [filename](#))
- [scrivi_docenti_a_contratto](#) (self, [df](#), [filename](#))
- [scrivi_ministeriali](#) (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__()`

```
__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()`

```
get_courses (
    self )
```

Ottiene un 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()`

```
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()`

```
get_professors (
    self )
```

Ottiene un 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()

```
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()

```
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()

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

Scrivi 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()

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

Scrivi 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()

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

Scrivi 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.

5.3.3.9 scrivi_ministeriali()

```
scrivi_ministeriali (
    self,
    df,
    filename )
```

Scrivi i dati del DataFrame in un file ASP (.lp), organizzandoli per sezioni con eliminazione di duplicati.
:param df: DataFrame contenente i dati da salvare. Deve contenere una colonna con il codice del corso,
il tipo di laurea secondo lo standard introdotto (ltss, lssm, ecc...), il numero di iscritti effettivi,
il numero massimo teorico (utilizzato per il calcolo della W)
: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

- [__init__](#) (self)
- [add_departments](#) (self, departments)
- [parse](#) (self)

Public Attributes

- [parser](#)

5.4.1 Constructor & Destructor Documentation

5.4.1.1 `__init__()`

```
__init__ (
    self )
```

Inizializza il parser per i dipartimenti.

5.4.2 Member Function Documentation

5.4.2.1 `add_departments()`

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

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

5.4.2.2 `parse()`

```
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

- namespace [main](#)

Functions

- [estrazione_dati_per_ssd](#) (path=[path_coperture](#))
 - [init](#) ()
 - [init_corsi](#) (filename)
 - [init_corsi_matricole](#) (filepathCorsi, filepathProf)
 - [init_matricole](#) (filename)
 - [main](#) ()
 - [main_old](#) ()
 - [process_ssd](#) (dataset_loader, ssd, failed_ssds)
 - [run_tests](#) ()
 - [scrittura_fatti](#) (filters=None, file_name=None, path=[path_coperture](#))
- FUNZIONI PRINCIPALI #####.*
- [test_estrazione_settori](#) ()
 - [test_filtra_per_colonne](#) ()
 - [test_filtra_per_valori](#) ()
 - [test_parser](#) ()
- TEST #####.*
- [unisci_file_per_settore](#) (path)

Variables

- str [path_coperture](#) = "dataset/coperture.xlsx"
- VARIABILI GLOBALI #####.*
- str [path_docenti](#) = "dataset/docenti.xlsx"
 - str [path_docenti_a_contratto](#) = "dataset/docenti_a_contratto.xlsx"
 - list [paths_immatricolati](#) = ["dataset/immatricolati/LT.xlsx", "dataset/immatricolati/LM.xlsx", "dataset/immatricolati/CU.xlsx"]

6.2 src/modules/__init__.py File Reference

6.3 src/modules/course_parser.py File Reference

Classes

- class [CourseParser](#)

Namespaces

- namespace [modules](#)
- namespace [modules.course_parser](#)

6.4 src/modules/dataset_loader.py File Reference

Classes

- class [DatasetLoader](#)

Namespaces

- namespace [modules](#)
- namespace [modules.dataset_loader](#)

6.5 src/modules/dataset_manager.py File Reference

Classes

- class [DatasetManager](#)

Namespaces

- namespace [modules](#)
- namespace [modules.dataset_manager](#)

6.6 src/modules/department_parser.py File Reference

Classes

- class [DepartmentParser](#)

Namespaces

- namespace [modules](#)
- namespace [modules.department_parser](#)

6.7 src/sanitize.py File Reference

Namespaces

- namespace [sanitize](#)

Functions

- [aggiorna_cod_tipo_corso](#) (row)
- [compute_extra_data](#) ()
- [compute_remained](#) ()
- [sanitize](#) ()
- [sanitize_codici_corso](#) (df)
- [sanitize_coperture](#) ()
- [sanitize_docenti](#) ()

Variables

- str [path_coperture](#) = "dataset/coperture.xlsx"
- str [path_coperture_contratti](#) = "dataset/docenti_a_contratto.xlsx"
- str [path_coperture_rimaste](#) = "dataset/insegnamenti_senza_docente.xlsx"
- str [path_docenti](#) = "dataset/docenti.xlsx"
- str [path_ns_coperture](#) = "dataset/originali/coperture.xlsx"
- str [path_ns_docenti](#) = "dataset/originali/docenti.xlsx"

Index

- `__init__`
 - `CourseParser`, 15
 - `DatasetLoader`, 16
 - `DatasetManager`, 19
 - `DepartmentParser`, 22
- `add_courses`
 - `CourseParser`, 15
- `add_departments`
 - `DepartmentParser`, 22
- `aggiorna_cod_tipo_corso`
 - `sanitize`, 12
- `compute_extra_data`
 - `sanitize`, 12
- `compute_remained`
 - `sanitize`, 12
- `CourseParser`, 15
 - `__init__`, 15
 - `add_courses`, 15
 - `parse`, 15
 - `parser`, 16
- `dataset_path`
 - `DatasetManager`, 21
- `DatasetLoader`, 16
 - `__init__`, 16
 - `df_to_dict`, 17
 - `filter_by_values`, 17
 - `get_values`, 17
 - `input_file_path`, 18
 - `save_to_file`, 18
- `DatasetManager`, 18
 - `__init__`, 19
 - `dataset_path`, 21
 - `get_courses`, 19
 - `get_departments`, 19
 - `get_professors`, 19
 - `get_sectors`, 19
 - `load_data`, 20
 - `scrivi_coperture`, 20
 - `scrivi_docenti`, 20
 - `scrivi_docenti_a_contratto`, 20
 - `scrivi_ministeriali`, 21
- `DepartmentParser`, 21
 - `__init__`, 22
 - `add_departments`, 22
 - `parse`, 22
 - `parser`, 22
- `df_to_dict`
 - `DatasetLoader`, 17
- `estrazione_dati_per_ssd`
 - `main`, 8
- `filter_by_values`
 - `DatasetLoader`, 17
- `get_courses`
 - `DatasetManager`, 19
- `get_departments`
 - `DatasetManager`, 19
- `get_professors`
 - `DatasetManager`, 19
- `get_sectors`
 - `DatasetManager`, 19
- `get_values`
 - `DatasetLoader`, 17
- `init`
 - `main`, 8
- `init_corsi`
 - `main`, 8
- `init_corsi_matricole`
 - `main`, 8
- `init_matricole`
 - `main`, 8
- `input_file_path`
 - `DatasetLoader`, 18
- `load_data`
 - `DatasetManager`, 20
- `main`, 7
 - `estrazione_dati_per_ssd`, 8
 - `init`, 8
 - `init_corsi`, 8
 - `init_corsi_matricole`, 8
 - `init_matricole`, 8
 - `main`, 8
 - `main_old`, 8
 - `path_coperture`, 10
 - `path_docenti`, 10
 - `path_docenti_a_contratto`, 10
 - `paths_immatricolati`, 10
 - `process_ssd`, 8
 - `run_tests`, 9
 - `scrittura_fatti`, 9
 - `test_estrazione_settori`, 9
 - `test_filtra_per_colonne`, 10
 - `test_filtra_per_valori`, 10

- test_parser, 10
- unisci_file_per_settore, 10
- main_old
 - main, 8
- modules, 11
- modules.course_parser, 11
- modules.dataset_loader, 11
- modules.dataset_manager, 11
- modules.department_parser, 11
- parse
 - CourseParser, 15
 - DepartmentParser, 22
- parser
 - CourseParser, 16
 - DepartmentParser, 22
- path_coperture
 - main, 10
 - sanitize, 13
- path_coperture_contratti
 - sanitize, 13
- path_coperture_rimaste
 - sanitize, 13
- path_docenti
 - main, 10
 - sanitize, 13
- path_docenti_a_contratto
 - main, 10
- path_ns_coperture
 - sanitize, 13
- path_ns_docenti
 - sanitize, 13
- paths_immatricolati
 - main, 10
- process_ssd
 - main, 8
- run_tests
 - main, 9
- sanitize, 12
 - aggiorna_cod_tipo_corso, 12
 - compute_extra_data, 12
 - compute_remained, 12
 - path_coperture, 13
 - path_coperture_contratti, 13
 - path_coperture_rimaste, 13
 - path_docenti, 13
 - path_ns_coperture, 13
 - path_ns_docenti, 13
 - sanitize, 12
 - sanitize_codici_corso, 12
 - sanitize_coperture, 12
 - sanitize_docenti, 13
- sanitize_codici_corso
 - sanitize, 12
- sanitize_coperture
 - sanitize, 12
- sanitize_docenti
 - sanitize, 13
- sanitize, 13
- save_to_file
 - DatasetLoader, 18
- scrittura_fatti
 - main, 9
- scrivi_coperture
 - DatasetManager, 20
- scrivi_docenti
 - DatasetManager, 20
- scrivi_docenti_a_contratto
 - DatasetManager, 20
- scrivi_ministeriali
 - DatasetManager, 21
- src/main.py, 23
- src/modules/__init__.py, 24
- src/modules/course_parser.py, 24
- src/modules/dataset_loader.py, 24
- src/modules/dataset_manager.py, 24
- src/modules/department_parser.py, 24
- src/sanitize.py, 25
- test_estrazione_settori
 - main, 9
- test_filtra_per_colonne
 - main, 10
- test_filtra_per_valori
 - main, 10
- test_parser
 - main, 10
- unisci_file_per_settore
 - main, 10