|  |
| --- |
| PS D:\Projects\github.com-YuriAbele\AIU\Python\L12\HW> & 'c:\Python313\python.exe' 'c:\Users\yuab\.vscode\extensions\ms-python.debugpy-2025.14.1-win32-x64\bundled\libs\debugpy\launcher' '56423' '--' 'D:\Projects\github.com-YuriAbele\  AIU\Python\L12\HW/main.py'    ==================================================  Started at: 2025-10-23 20:58:36  ####################################################################################################    Ensuring filesystem structure:START  ['project\_root', 'project\_root/logs', 'project\_root/data', 'project\_root/data/raw', 'project\_root/data/processed', 'project\_root/output', 'project\_root/backups']    --> Ensuring directory exists: project\_root  --> Ensuring directory exists: project\_root/logs  --> Ensuring directory exists: project\_root/data  --> Ensuring directory exists: project\_root/data/raw  --> Ensuring directory exists: project\_root/data/processed  --> Ensuring directory exists: project\_root/output  --> Ensuring directory exists: project\_root/backups  Ensuring filesystem structure:END    ####################################################################################################    Cleaning encoding example files:START  --> Delete the "./project\_root/data/raw/example\_ASCII.txt" file.  --> Delete the "./project\_root/data/raw/example\_ISO-8859-5.txt" file.  --> Delete the "./project\_root/data/raw/example\_UTF16.txt" file.  --> Delete the "./project\_root/data/raw/example\_UTF8.txt" file.  --> Delete the "./project\_root/data/raw/example\_Windows-1251.txt" file.  --> Delete the "./project\_root/data/processed/example\_ASCII\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_ISO-8859-5\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_UTF16\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_UTF8\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_Windows-1251\_processed.txt" file.  Files before: 10, Files after: 0  Cleaning encoding example files:END    Generating encoding example files:START  --> Create or replace the "./project\_root/data/raw/example\_UTF8.txt" file with "UTF-8" encoding.  --> Create or replace the "./project\_root/data/raw/example\_UTF16.txt" file with "UTF-16" encoding.  --> Create or replace the "./project\_root/data/raw/example\_ASCII.txt" file with "ASCII" encoding.  --> Create or replace the "./project\_root/data/raw/example\_ISO-8859-5.txt" file with "ISO-8859-5" encoding.  --> Create or replace the "./project\_root/data/raw/example\_Windows-1251.txt" file with "Windows-1251" encoding.  Generating encoding example files:END    ####################################################################################################    Reading encoding example files:START  --> Processing file: example\_UTF8.txt  --> Read the "./project\_root/data/raw/example\_UTF8.txt" file.  --> Detected encoding for "./project\_root/data/raw/example\_UTF8.txt": utf-8 with confidence 0.99  --> Write the processed content to "./project\_root/data/processed/example\_UTF8\_processed.txt" file.  --> Processing file: example\_UTF16.txt  --> Read the "./project\_root/data/raw/example\_UTF16.txt" file.  --> Detected encoding for "./project\_root/data/raw/example\_UTF16.txt": UTF-16 with confidence 1.00  --> Write the processed content to "./project\_root/data/processed/example\_UTF16\_processed.txt" file.  --> Processing file: example\_ASCII.txt  --> Read the "./project\_root/data/raw/example\_ASCII.txt" file.  --> Detected encoding for "./project\_root/data/raw/example\_ASCII.txt": ascii with confidence 1.00  --> Write the processed content to "./project\_root/data/processed/example\_ASCII\_processed.txt" file.  --> Processing file: example\_ISO-8859-5.txt  --> Read the "./project\_root/data/raw/example\_ISO-8859-5.txt" file.  --> Detected encoding for "./project\_root/data/raw/example\_ISO-8859-5.txt": ISO-8859-5 with confidence 0.98  --> Write the processed content to "./project\_root/data/processed/example\_ISO-8859-5\_processed.txt" file.  --> Processing file: example\_Windows-1251.txt  --> Read the "./project\_root/data/raw/example\_Windows-1251.txt" file.  --> Detected encoding for "./project\_root/data/raw/example\_Windows-1251.txt": windows-1251 with confidence 0.95  --> Write the processed content to "./project\_root/data/processed/example\_Windows-1251\_processed.txt" file.  Reading encoding example files:END    ####################################################################################################    Read many files:START  Process file\_name "example\_UTF8.txt":START  --> Read 679 characters from "./project\_root/data/raw/example\_UTF8.txt" file.  --> Read 679 characters from "./project\_root/data/processed/example\_UTF8\_processed.txt" file.  Process file\_name "example\_UTF8.txt":END  Process file\_name "example\_UTF16.txt":START  --> Read 679 characters from "./project\_root/data/raw/example\_UTF16.txt" file.  --> Read 679 characters from "./project\_root/data/processed/example\_UTF16\_processed.txt" file.  Process file\_name "example\_UTF16.txt":END  Process file\_name "example\_ASCII.txt":START  --> Read 679 characters from "./project\_root/data/raw/example\_ASCII.txt" file.  --> Read 679 characters from "./project\_root/data/processed/example\_ASCII\_processed.txt" file.  Process file\_name "example\_ASCII.txt":END  Process file\_name "example\_ISO-8859-5.txt":START  --> Read 679 characters from "./project\_root/data/raw/example\_ISO-8859-5.txt" file.  --> Read 679 characters from "./project\_root/data/processed/example\_ISO-8859-5\_processed.txt" file.  Process file\_name "example\_ISO-8859-5.txt":END  Process file\_name "example\_Windows-1251.txt":START  --> Read 679 characters from "./project\_root/data/raw/example\_Windows-1251.txt" file.  --> Read 679 characters from "./project\_root/data/processed/example\_Windows-1251\_processed.txt" file.  Process file\_name "example\_Windows-1251.txt":END  Read many files:END    Convert a list to a JSON array:START  --> Got 23823 characters  Convert a list to a JSON array:END    Saving text to file:START  --> Saved 23823 bytes to "./project\_root/output/processed\_data.json" file.  Saving text to file:END    ####################################################################################################    Collecting tree info for directory:START  project\_root/data    Calculating file hash:START  --> File: project\_root/data/processed/example\_ASCII\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 9f6c333dbc4c3ebb437b136ccbc6e1b3630c14b5a2d567f0e0464e02b48e0a8b    Calculating file hash:START  --> File: project\_root/data/processed/example\_ISO-8859-5\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: f2d1ea8f7036ff47c406b6fa25096a5fbab088c6fa5c8cb2744dc3f1eb4fd35a    Calculating file hash:START  --> File: project\_root/data/processed/example\_UTF16\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 97ccab3c3edceb5f98ebb758907fb14d4af5cdf7eb7da7e9e9bc180ded251bd9    Calculating file hash:START  --> File: project\_root/data/processed/example\_UTF8\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 69fc65e03e6feacccb4d76b4afb650a0165aba84775f02cea2c1c15eb850e875    Calculating file hash:START  --> File: project\_root/data/processed/example\_Windows-1251\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: a7043c61d84ab76119d566285190cb878d6a499bcf913747e7644438757f4b38    Calculating file hash:START  --> File: project\_root/data/raw/example\_ASCII.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 0eef845262be8ebcb28d66663522a3482ec2b4c090e6cdd94ef40017784af3f3    Calculating file hash:START  --> File: project\_root/data/raw/example\_ISO-8859-5.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 3e6bf34f299a25adf6417fd01bb835508653d81a266c6a710c44e27af5907fbf    Calculating file hash:START  --> File: project\_root/data/raw/example\_UTF16.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: aeb6b2f78be002c682cec70e10abd0f259d31f41973fa605fa162ff71cb14ff5    Calculating file hash:START  --> File: project\_root/data/raw/example\_UTF8.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: e20eac539ad0d286fbf7800013deb4213ad1ef4921f26f8f185b3db7be088fbe    Calculating file hash:START  --> File: project\_root/data/raw/example\_Windows-1251.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 07d0a35e523520ab2f1668eb65a1fe7cc5f5fc426b503630bd84c24629fb6821  Collecting tree info for directory:END    Creating backup for "project\_root/data":START  --> Backup created at "./project\_root/backups/backup\_20251023.zip" (Size: 6573 bytes)  Creating backup for "project\_root/data":END    ####################################################################################################    Cleaning encoding example files:START  --> Delete the "./project\_root/data/raw/example\_ASCII.txt" file.  --> Delete the "./project\_root/data/raw/example\_ISO-8859-5.txt" file.  --> Delete the "./project\_root/data/raw/example\_UTF16.txt" file.  --> Delete the "./project\_root/data/raw/example\_UTF8.txt" file.  --> Delete the "./project\_root/data/raw/example\_Windows-1251.txt" file.  --> Delete the "./project\_root/data/processed/example\_ASCII\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_ISO-8859-5\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_UTF16\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_UTF8\_processed.txt" file.  --> Delete the "./project\_root/data/processed/example\_Windows-1251\_processed.txt" file.  Files before: 10, Files after: 0  Cleaning encoding example files:END    Get full path of latest backup file in "project\_root/backups":START  --> Latest backup file found: "./project\_root/backups/backup\_20251023.zip"  Get full path of latest backup file in "project\_root/backups":END    Unpacking backup "./project\_root/backups/backup\_20251023.zip" to "project\_root/data":START  --> Restored files:  project\_root/data/  ├── processed  │ ├── example\_ASCII\_processed.txt  │ ├── example\_ISO-8859-5\_processed.txt  │ ├── example\_UTF16\_processed.txt  │ ├── example\_UTF8\_processed.txt  │ └── example\_Windows-1251\_processed.txt  └── raw  ├── example\_ASCII.txt  ├── example\_ISO-8859-5.txt  ├── example\_UTF16.txt  ├── example\_UTF8.txt  └── example\_Windows-1251.txt  Unpacking backup "./project\_root/backups/backup\_20251023.zip" to "project\_root/data":END    Collecting tree info for directory:START  project\_root/data    Calculating file hash:START  --> File: project\_root/data/processed/example\_ASCII\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 9f6c333dbc4c3ebb437b136ccbc6e1b3630c14b5a2d567f0e0464e02b48e0a8b    Calculating file hash:START  --> File: project\_root/data/processed/example\_ISO-8859-5\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: f2d1ea8f7036ff47c406b6fa25096a5fbab088c6fa5c8cb2744dc3f1eb4fd35a    Calculating file hash:START  --> File: project\_root/data/processed/example\_UTF16\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 97ccab3c3edceb5f98ebb758907fb14d4af5cdf7eb7da7e9e9bc180ded251bd9    Calculating file hash:START  --> File: project\_root/data/processed/example\_UTF8\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 69fc65e03e6feacccb4d76b4afb650a0165aba84775f02cea2c1c15eb850e875    Calculating file hash:START  --> File: project\_root/data/processed/example\_Windows-1251\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: a7043c61d84ab76119d566285190cb878d6a499bcf913747e7644438757f4b38    Calculating file hash:START  --> File: project\_root/data/raw/example\_ASCII.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 0eef845262be8ebcb28d66663522a3482ec2b4c090e6cdd94ef40017784af3f3    Calculating file hash:START  --> File: project\_root/data/raw/example\_ISO-8859-5.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 3e6bf34f299a25adf6417fd01bb835508653d81a266c6a710c44e27af5907fbf    Calculating file hash:START  --> File: project\_root/data/raw/example\_UTF16.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: aeb6b2f78be002c682cec70e10abd0f259d31f41973fa605fa162ff71cb14ff5    Calculating file hash:START  --> File: project\_root/data/raw/example\_UTF8.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: e20eac539ad0d286fbf7800013deb4213ad1ef4921f26f8f185b3db7be088fbe    Calculating file hash:START  --> File: project\_root/data/raw/example\_Windows-1251.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 07d0a35e523520ab2f1668eb65a1fe7cc5f5fc426b503630bd84c24629fb6821  Collecting tree info for directory:END    Compare FileInfo lists:START  --> Both lists have 10 items  --> Both lists are identical  Compare FileInfo lists:END  Are the original and restored file info lists equal? - True    ####################################################################################################    Collecting tree info for directory:START  project\_root/data/processed    Calculating file hash:START  --> File: project\_root/data/processed/example\_ASCII\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 9f6c333dbc4c3ebb437b136ccbc6e1b3630c14b5a2d567f0e0464e02b48e0a8b    Calculating file hash:START  --> File: project\_root/data/processed/example\_ISO-8859-5\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: f2d1ea8f7036ff47c406b6fa25096a5fbab088c6fa5c8cb2744dc3f1eb4fd35a    Calculating file hash:START  --> File: project\_root/data/processed/example\_UTF16\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 97ccab3c3edceb5f98ebb758907fb14d4af5cdf7eb7da7e9e9bc180ded251bd9    Calculating file hash:START  --> File: project\_root/data/processed/example\_UTF8\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: 69fc65e03e6feacccb4d76b4afb650a0165aba84775f02cea2c1c15eb850e875    Calculating file hash:START  --> File: project\_root/data/processed/example\_Windows-1251\_processed.txt  --> Algorithm: sha256  --> Block size: 65536 bytes  Calculating file hash:END  Hash: a7043c61d84ab76119d566285190cb878d6a499bcf913747e7644438757f4b38  Collecting tree info for directory:END  FileInfo(full\_path='project\_root/data/processed/example\_ASCII\_processed.txt', file\_name='example\_ASCII\_processed.txt', size=691, last\_modified\_at='2025-10-23 20:58:36', hash='9f6c333dbc4c3ebb437b136ccbc6e1b3630c14b5a2d567f0e0464e02b48  e0a8b')  FileInfo(full\_path='project\_root/data/processed/example\_ISO-8859-5\_processed.txt', file\_name='example\_ISO-8859-5\_processed.txt', size=691, last\_modified\_at='2025-10-23 20:58:36', hash='f2d1ea8f7036ff47c406b6fa25096a5fbab088c6fa5c8cb27  44dc3f1eb4fd35a')  FileInfo(full\_path='project\_root/data/processed/example\_UTF16\_processed.txt', file\_name='example\_UTF16\_processed.txt', size=1384, last\_modified\_at='2025-10-23 20:58:36', hash='97ccab3c3edceb5f98ebb758907fb14d4af5cdf7eb7da7e9e9bc180ded  251bd9')  FileInfo(full\_path='project\_root/data/processed/example\_UTF8\_processed.txt', file\_name='example\_UTF8\_processed.txt', size=1103, last\_modified\_at='2025-10-23 20:58:36', hash='69fc65e03e6feacccb4d76b4afb650a0165aba84775f02cea2c1c15eb850  e875')  FileInfo(full\_path='project\_root/data/processed/example\_Windows-1251\_processed.txt', file\_name='example\_Windows-1251\_processed.txt', size=691, last\_modified\_at='2025-10-23 20:58:36', hash='a7043c61d84ab76119d566285190cb878d6a499bcf913  747e7644438757f4b38')    Convert a list to a JSON array:START  --> Got 1526 characters  Convert a list to a JSON array:END    Saving text to file:START  --> Saved 1526 bytes to "./project\_root/output/processed\_data\_file\_infos.json" file.  Saving text to file:END    Printing file contents:START  File: ./project\_root/output/processed\_data\_file\_infos.json  Contents of the file infos JSON output file:  [  {  "file\_name": "example\_ASCII\_processed.txt",  "full\_path": "project\_root/data/processed/example\_ASCII\_processed.txt",  "size": 691,  "last\_modified\_at": "2025-10-23 20:58:36",  "hash": "9f6c333dbc4c3ebb437b136ccbc6e1b3630c14b5a2d567f0e0464e02b48e0a8b"  },  {  "file\_name": "example\_ISO-8859-5\_processed.txt",  "full\_path": "project\_root/data/processed/example\_ISO-8859-5\_processed.txt",  "size": 691,  "last\_modified\_at": "2025-10-23 20:58:36",  "hash": "f2d1ea8f7036ff47c406b6fa25096a5fbab088c6fa5c8cb2744dc3f1eb4fd35a"  },  {  "file\_name": "example\_UTF16\_processed.txt",  "full\_path": "project\_root/data/processed/example\_UTF16\_processed.txt",  "size": 1384,  "last\_modified\_at": "2025-10-23 20:58:36",  "hash": "97ccab3c3edceb5f98ebb758907fb14d4af5cdf7eb7da7e9e9bc180ded251bd9"  },  {  "file\_name": "example\_UTF8\_processed.txt",  "full\_path": "project\_root/data/processed/example\_UTF8\_processed.txt",  "size": 1103,  "last\_modified\_at": "2025-10-23 20:58:36",  "hash": "69fc65e03e6feacccb4d76b4afb650a0165aba84775f02cea2c1c15eb850e875"  },  {  "file\_name": "example\_Windows-1251\_processed.txt",  "full\_path": "project\_root/data/processed/example\_Windows-1251\_processed.txt",  "size": 691,  "last\_modified\_at": "2025-10-23 20:58:36",  "hash": "a7043c61d84ab76119d566285190cb878d6a499bcf913747e7644438757f4b38"  }  ]  Printing file contents:END    Convert a JSON array to a list:START  --> Got 5 FileInfo objects  Convert a JSON array to a list:END    Compare FileInfo lists:START  --> Both lists have 5 items  --> Both lists are identical  Compare FileInfo lists:END  Are the original and deserialized file info lists equal? - True    ####################################################################################################    Loading file contents as string:START  --> File: ./project\_root/output/processed\_data\_file\_infos.json  --> Encoding: utf-8  --> Read 1526 characters from "./project\_root/output/processed\_data\_file\_infos.json" file.  Loading file contents as string:END    Load JSON schema from file:START  --> Loaded schema from "./schemas/json-schema-file\_infos\_list.json"  {  "$schema": "http://json-schema.org/draft-07/schema#",  "type": "array",  "description": "",  "minItems": 1,  "uniqueItems": true,  "items": {  "type": "object",  "required": [  "file\_name",  "full\_path",  "size",  "last\_modified\_at",  "hash"  ],  "additionalProperties": false,  "properties": {  "file\_name": {  "type": "string",  "minLength": 1  },  "full\_path": {  "type": "string",  "minLength": 1  },  "size": {  "type": "number"  },  "last\_modified\_at": {  "type": "string",  "minLength": 1  },  "hash": {  "type": "string",  "minLength": 1  }  }  }  }  Load JSON schema from file:END    Validate JSON against schema:START  --> JSON is valid against the schema.  Validate JSON against schema:END  JSON Schema validation: SUCCESS    ####################################################################################################  Finished at: 2025-10-23 20:58:36  ==================================================    PS D:\Projects\github.com-YuriAbele\AIU\Python\L12\HW> |