

Third homework for Architecture.

Variant:

- *Artifact 10: Well of wisdom*
- *Function 10: Straight insertion sort*

Files:

- **main.py(1,34KB)**: main function which reads, writes and catches exceptions coming from the container...
- **container.py(2,52KB)**: container with all the functions for containing inputted objects .
- **base_lines.py(476B)**: basic well of wisdom structure with all the functions.
- **aphorism.py(305B)**: aphorism structure with all the functions.
- **proverb.py(307B)**: proverb structure with all the functions.
- **riddle.py(301B)**: riddle structure with all the functions.
- **extension.py(43B)**: contains imports for main.py.

Run time:

- 2 elements - 00.027999 seconds
- 20 elements - 00.271038 seconds
- 10000 elements - 02:32.941722 minutes

Run time is probably way longer than in C++ because of language optimality, but all in all generating 10000 elements with long strings and comparing them in an suboptimal sort takes lot of time.

Command line input guide:

1) Write *python3 main.py -f [inputFileName].txt [outputFileName].txt [sortedOutputFileName].txt* for file input. 2) Write *python3 main.py -n [number of wells of wisdom] [outputFileName].txt [sortedOutputFileName].txt* for random input generation.

File input guide:

You need to input a couple of objects according to this template:

1) Input type: 1 for aphorism, 2 for proverb, 3 for riddle. 2) Input text of artifact: line of chars with length 10.000 or less. 3) Input meta text for artifact (author name, country or riddle answer): line of chars with length 10.000 or less.

Tests:

There are 5 tests stored in tests directory, which could be used to understand if the program runs properly and tests could be autogenerated using command line. They are named after what they are showcasing.

Memory:

Classes function display:

Class	Function name	Description	Memory influence
Container	init	def exemplar function	lines: list
Container	str	def exemplar function	lines: list
Container	input_from_file	def exemplar function	lines: list, path: str, inp: str, file: file
Container	random_input	def exemplar function	lines: list, count: int, inp: int
Container	sort	def exemplar function	lines: list
Container	out_to_file	def exemplar function	path: str, prefix: str, file: file
Container	generate_string	def static function	s_builder: list

Class	Function name	Description	Memory influence
BaseLines	init	def exemplar function	line: str, comparable: float
BaseLines	__compute_comparable	def exemplar function	punctutaion_count: int, line: str, length: int
Aphorism	init	def exemplar function	line: str, comparable: float, author: str
Aphorism	str	def exemplar function	line: str, comparable: float, author: str
Riddle	init	def exemplar function	line: str, comparable: float, answer: str
Riddle	str	def exemplar function	line: str, comparable: float, answer: str
Proverb	init	def exemplar function	line: str, comparable: float, country: str
Proverb	str	def exemplar function	line: str, comparable: float, country: str

Modules:

- Implementation modules count: 7