Porter, Karli
Sotomayor, Tyler

COP 2930 | Select Topics in Computer Programming - Python
Valencia College (Spring 2023)
Dennis Hunchuck

Submitted: 19 March 2023

Design Tools, UML Diagrams, and Visual Manifestations

- = attribute
[m] = method
[f] = function

Class UML Diagrams

Automobile	
-vin: str	17 char alphanumeric string
-make: str	Make of the auto, sentence case format
-model: str	Model of the auto, sentence case format
-year: int	Year auto was made
-price: float	Selling price of auto
[m] get_vin(vin: str)	Return vehicle ID number of auto
[m] get_make(make: str)	Return make of auto
[m] get_model(model: str)	Return model of auto
[m] get_year(year: int)	Return year of auto
[m] get_price(price: float)	Return price of auto
[m] set_vin(vin: str)	Validates and sets string as 17 char unique alphanumeric string
[m] set_make(make: str)	Validates and sets string in sentence case format
[m] set_model(model: str)	Validates and sets string in sentence case format
[m] set_year(year: int)	Validates and sets year as integer between 1900 and 2024
[m] set_price(price: float)	Validates and sets float as comma separated float between 0 and 200,000.

Automobile	
[m] __str__() → str	Formatted string for automobile object's attributes. Padded and justified so decimals line up.

AutomobileInventory	
- automobiles: List[Automobile]	Initialized in __init__ method
[m] add_auto(auto: Automobile)	Adds automobile to inventory.
[m] display_autos()	Shows entire list of automobiles in order they were added to inventory.
[m] get_stats()	Shows statistics (number of autos, most/least expensive, average price)
[m] save_data()	Saves current inventory to JSON file.
[m] load_data()	Loads inventory from JSON file.
[m] clear_data()	Clears data from automobiles list and saves the state to JSON file.

Module UML Diagrams

sample_data.py	
[f] generate_random_autos()	Generates sample data from a list of makes and models.

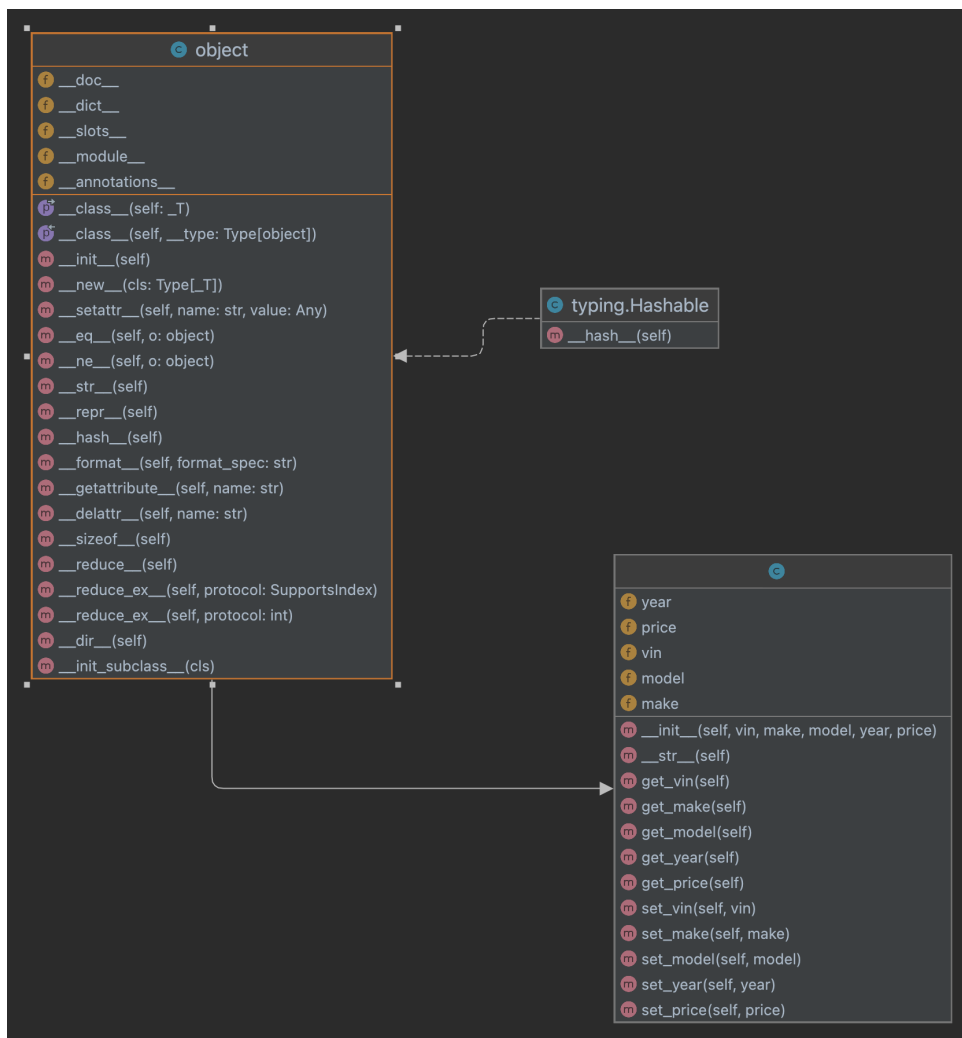| menu_actions.py	
[f] get_user_selection()	The main menu
[f] add_auto()	User enters auto information into inventory
[f] add_sample_data()	Adds a select number of randomly generated sample data to the inventory
[f] display_autos()	Shows a list of autos and their details in the order they were added to the inventory
[f] show_stats()	Shows statistics (number of autos, most/least expensive, average price)
[f] save_data()	Saves data as a JSON file to a local file. "Data saved successfully."

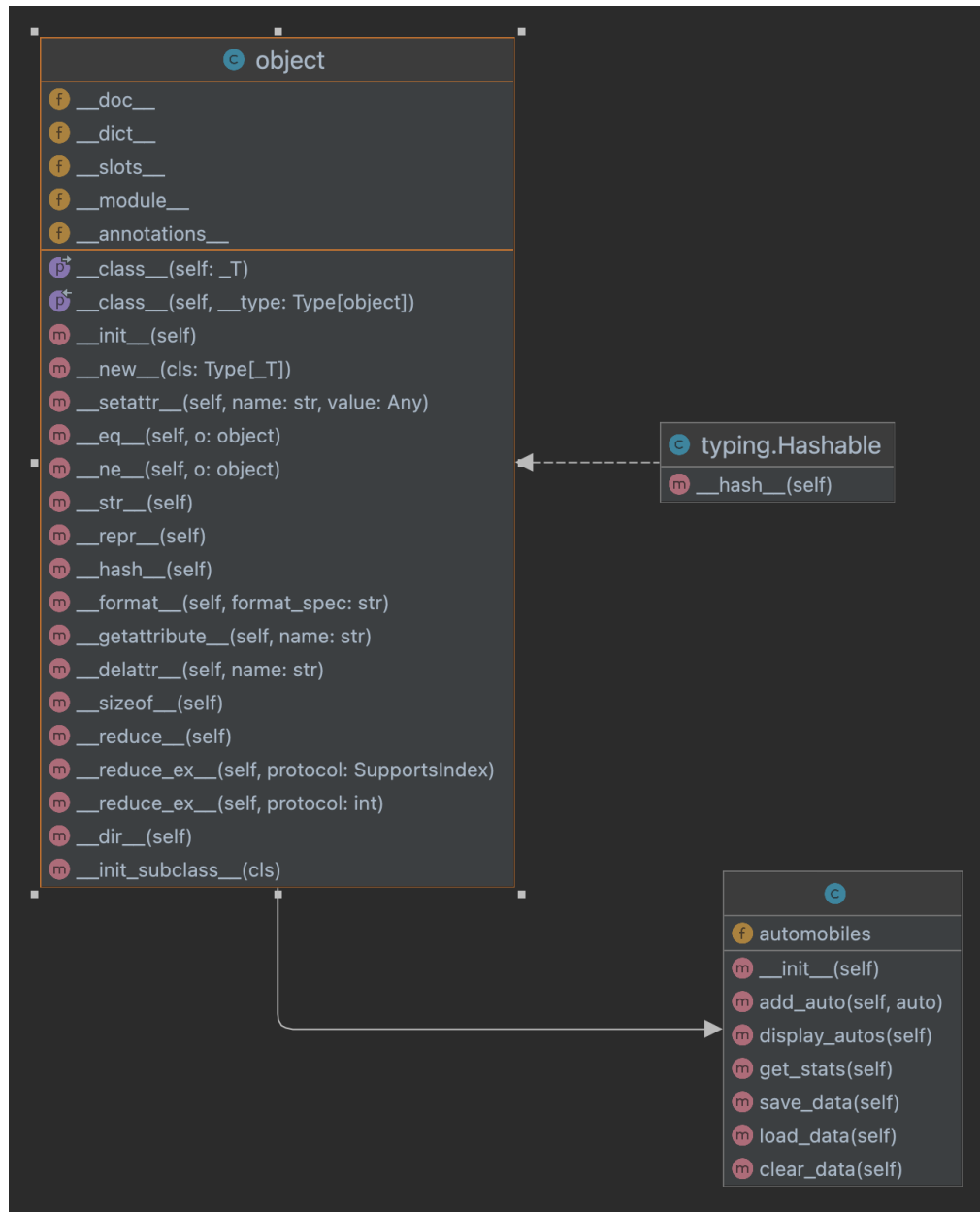| menu_actions.py	
[f] load_data()	JSON saved data can be referenced in between sessions. "Data file not found" / "Data loaded successfully"
[f] clear_data	"Data cleared successfully"
[f] quit_program	"Goodbye!"

main.py	
[f] main()	

PyCharm generated UML Class Diagrams:

Class: Automobile

Class: AutomobileInventory

object

- f __doc__
- f __dict__
- f __slots__
- f __module__
- f __annotations__
- P __class__(self: _T)
- P __class__(self, __type: Type[object])
- m __init__(self)
- m __new__(cls: Type[_T])
- m __setattr__(self, name: str, value: Any)
- m __eq__(self, o: object)
- m __ne__(self, o: object)
- m __str__(self)
- m __repr__(self)
- m __hash__(self)
- m __format__(self, format_spec: str)
- m __getattribute__(self, name: str)
- m __delattr__(self, name: str)
- m __sizeof__(self)
- m __reduce__(self)
- m __reduce_ex__(self, protocol: SupportsIndex)
- m __reduce_ex__(self, protocol: int)
- m __dir__(self)
- m __init_subclass__(cls)

typing.Hashable

- m __hash__(self)

c

- f automobiles
- m __init__(self)
- m add_auto(self, auto)
- m display_autos(self)
- m get_stats(self)
- m save_data(self)
- m load_data(self)
- m clear_data(self)

Flowchart

Visualizes the Main Menu