Classes

1. RentalUILogic

Handles the core rental logic.

Attributes:

shop: A Store object representing the rental shop.

customer_rentals: Dictionary storing active rentals keyed by customer ID.

daily_ski_rentals: Counter for ski rentals per day.

daily_snowboard_rentals: Counter for snowboard rentals per day.

revenu: Total revenue collected for the day.

Methods:

set_shop(skis, snowboards): Initializes the store with a given inventory.

estimate(...): Returns a formatted rental estimate.

is_inventory_sufficient(...): Checks equipment availability.

is_customer_id_valid(...): Ensures customer ID is unique for active rentals.

new_rental(...): Registers a new rental with equipment, rental type, and discount.

return_rental(...): Handles rental return and billing logic, updates inventory and revenue.

2. RentalUI

User Interface class to manage input/output and control flow.

Attributes:

isDebug: If true, enables manual datetime input for testing.

logic: Instance of RentalUILogic.

Methods:

BuildStore(): Prompts user to initialize store inventory.

wait(): Pauses the program.

yes_no(question): Returns a boolean from a y/n prompt.

validate_int_input(prompt): Ensures a positive integer input.

validate_rental_type(): Validates rental type selection.

get_time_input(): Returns current time, or user input in debug mode.

MainMenu(): Displays the main navigation menu.

new_customer_rental(): Orchestrates a new customer rental.

rental_return(): Handles the return process and prints invoice.

show_inventory(): Displays current inventory levels.

end_of_day(): Shows the day's summary and exits the application.

Changes to the Code:

1. Fixed variable reference in `RequestEquipment` method:
    - In `classes-1.py`: Uses undeclared variables `if Skis > self.storeName.CurrentSki or Snowboards > self.storeName.CurrentSnow:`
    - In `classes.py`: Fixed to use instance variables `if self.Skis > self.storeName.CurrentSki or self.Snowboards > self.storeName.CurrentSnow:`
2. Updated `returnItem` method comment:
    - In `classes-1.py`: "Allows customers to return their bikes to the rental shop."
    - In `classes.py`: "Allows customers to return their items to the rental shop."
3. Enhanced `calculateRentalCost` method:
    - In `classes.py`: Added optional `return_time` parameter with default value of `None`
    - In `classes.py`: Uses the provided return time or defaults to current time

    - In `classes.py`: Added `max()` functions to ensure minimum duration of 1 for calculations
4. Changed `finalCost` method:
    - In `classes.py`: The method now returns `dblFinalCost`
5. Commented out test code:
    - In `classes.py`: All the test code at the end (customer creation, store creation, rental operations) is commented out
    - In `classes-1.py`: The test code is active and will run when the file is executed

The most significant improvements in `classes.py` are the fixed variable references, the ability to specify a return time for rental cost calculations, and ensuring minimum rental periods of at least 1 unit (hour/day/week).