README - Customer Reward System

Synopsis

Customer Reward System is a Ruby 3 implementation of the *Rule Engine Pattern* with some differences from the textbook version:

- Rules can be dynamically added to the rule engine.
- Rules can be dynamically removed from the rule engine.
- Rules are evaluated in reverse chronological order of when they were added to the rule engine.
- The rule engine caters for no rules matching at all.
- When a rule matches the remaining rules are not used in the rule engine.

It is written using SOLID principles with testability, extensibility and performance in mind. The source code is all found in *lib* and uses only the Ruby 3 standard library, making no use of external gems. The test code is all found in *spec* and uses RSpec. All Ruby 3 files written for this project adhere to the code convention of StandardRB.

Quick Start

In the directory *lib* type `ruby customer_rewards_analyzer.rb` to see output to STDOUT based on the input test data in the file.

Caveat

Class *CustomerRewardsAnalyzer* does ***not*** process the list of results it produces for the purchase data input it analyzes. For example, it may record a customer's next purchase will be free, but it will not state that as the reward for their next purchase. The results array it gives needs further processing downstream to ensure the correct rewards are ultimately received by customers for their purchases.

Files

Base Directory

- *.rspec* - Generated automatically by command `rspec --init`

- *Gemfile* - The project gem file

- *Gemfile.lock* - The frozen gem file

- *README.html* - The self-contained HTML5 version of this file

- *README.md* - The markdown version of this file
- *README.pdf* - The PDF version of this file
- *Specification.pdf* - A specification document for this project

lib Directory

- *customer_context.rb* - The implementation of *CustomerContext* which is the key class the rules engine works with
- *customer.rb* - The implementation of a *Customer* class for all static data related to a customer
- *customer_rewards_analyzer.rb* - The implementation of class *CustomerRewardsAnalyzer* which takes a rule engine and a list of customer purchases to create a list of any rewards for those purchases
- *interface_reward.rb* - Interface *InterfaceReward* defines the interface for all concrete reward classes
- *interface_rule_engine.rb* - Interface *InterfaceRuleEngine* defines the interface for all concrete rule engine classes
- *interface_rule.rb* - Interface *InterfaceRule* defines the interface for all concrete rule classes
- *purchase.rb* - The *Purchase* class holds all data related to a purchase and is stored in *CustomerContext* objects
- *reward_next_purchase_free.rb* - Concrete reward class *RewardNextPurchaseFree* represents a reward of the next purchase of a customer being free
- *reward_none.rb* - Concrete reward class RewardNone is assigned when the rule engine finds no rules apply to a purchase
- *reward_percent_off_next_purchase.rb* - Concrete reward class *RewardPercentOffNextPurchase* represents some percentage discount on the price of a subsequent purchase
- *reward_star_wars.rb* - Concrete reward class *RewardStarWars* is the reward for a purchase on May 4th
- *rule_anniversary.rb* - Concrete rule class *RuleAnniversary* checks the time of a purchase against a given anniversary day
- *rule_buy_limit.rb* - Concrete rule class *RuleBuyLimit* activates when the amount purchased in some currency is at least a given threshold

- *rule_engine_dynamic_choose_latest.rb* - Concrete rule engine class *RuleEngineDynamicChooseLatest* takes a *CustomerContext* object and determines which rules apply to it

- *rule_null.rb* - Concrete rule class *RuleNull* is valid only when all other rules in the rule engine do not apply

- *rule_time_limit.rb* - Concrete rule class *RuleTimeLimit* checks if a customer made another purchase within some time threshold in seconds since their previous purchase

spec Directory

- *customer_context_spec.rb* - RSpec test file for class *CustomerContext*

- *customer_rewards_analyzer_spec.rb* - RSpec test file for class *CustomerRewardsAnalyzer*

- *customer_spec.rb* - RSpec test file for class *Customer*

- *interface_reward_spec.rb* - RSpec test file for interface *InterfaceReward*

- *interface_rule_engine_spec.rb* - RSpec test file for interface *InterfaceRuleEngine*

- *interface_rule_spec.rb* - RSpec test file for interface *InterfaceRule*

- *purchase_spec.rb* - RSpec test file for class *Purchase*

- *reward_next_purchase_free_spec.rb* - RSpec test file for class *RewardNextPurchaseFree*

- *reward_none_spec.rb* - RSpec test file for class *RewardNone*

- *reward_percent_off_next_purchase_spec.rb* - RSpec test file for class *RewardPercentOffNextPurchase*

- *reward_star_wars_spec.rb* - RSpec test file for class *RewardStarWars*

- *rule_anniversary_spec.rb* - RSpec test file for class *RuleAnniversary*

- *rule_buy_limit_spec.rb* - RSpec test file for class *RuleBuyLimit*

- *rule_engine_dynamic_choose_latest_spec.rb* - RSpec test file for class *RuleEngineDynamicChooseLatest*

- *rule_null_spec.rb* - RSpec test file for class *RuleNull*

- *rule_time_limit_spec.rb* - RSpec test file for class *RuleTimeLimit*

- *spec_helper.rb* - Generated automatically by command `rspec --init`

Ruby version

Ruby version 3.0.2 was used in creating this project.

System Dependencies

- The source code in *lib* directory is in Ruby 3 relying on the standard library only.
- Test code is found in *spec* directory and that requires RSpec to be installed in order to run.
- Ruby files written for this project adhere to the coding standard of StandardRB. This can be installed by the command `gem install standardrb` if it is not present.

Configuration

No specific configuration is needed.

Testing

Installing RSpec

The command `gem install rspec` will install the RSpec gem.

Running Tests

In the project base directory command `rspec` will run all the tests in *spec* directory that are based on the source code in *lib* directory.

Notes

- This project is not to be mistaken as a production-ready system.
- The test suite in *spec* reflects all of the information found in document *Specification.pdf* .
- The RSpec test files in *spec* show the recommended way to use the classes and interfaces of this project.