ReusabiliTokenSimulator

v1

Generated by Doxygen 1.8.14

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 app_reusabilityToken_simulator Namespace Reference

Functions

- def **setup_args** ()
- def **run_simulator** ()

Variables

- int **num_iterations** = 100
- int **num_customers** = 3
- int **num_shops** = 2

4.1.1 Detailed Description

```
@package app_reusabilityToken_simulator
Simulates a market where reusability tokens are at work

For usage: python app_reusabilityToken_simulator.py --help
```

4.2 Customer Namespace Reference

Classes

- class BadCustomer
- class Customer
- class GoodCustomer
- class NeutralCustomer

4.2.1 Detailed Description

```
@package Customer
Implementation of a generic Customer
```

4.3 Shop Namespace Reference

Classes

- class Shop

4.3.1 Detailed Description

```
@package Shop
Implementation of a shop
```

4.4 ShopListOracle Namespace Reference

Classes

- class ShopListOracle

4.4.1 Detailed Description

```
@package ShopListOracle
Implementation of a simple shop list oracle
```

4.5 SimulationEngine Namespace Reference

Classes

- class SimulationEngine

4.5.1 Detailed Description

```
@package SimulationEngine
Implementation of a simulation Engine
```

4.6 SimulationTimeOracle Namespace Reference

Classes

- class SimulationTimeOracle

4.6.1 Detailed Description

```
@package SimulationTimeOracle
Implementation of a timing oracle
```

4.7 SmartContract Namespace Reference

Classes

- class SmartContract

4.7.1 Detailed Description

```
@package SmartContract
Implementation of a smart contract
```

4.8 Visualization Namespace Reference

Functions

- def visualize_function (values, name, color, ax=None)
- def visualize_market (smart_contract, customer_list, shop_list, ax_cus=None, ax_shop=None, ax_cp=None, ax_ca=None)
- def diminishing_returns (max_val, b1, num_samples=100)

4.8.1 Detailed Description

```
@package Visualization
Implementation of a few visualizers
```

4.8.2 Function Documentation

4.8.2.1 diminishing_returns()

```
def Visualization.diminishing_returns (
              max_val,
              b1,
              num_samples = 100 )
```

```
Test function that generates values from a diminishing returns function
:param max_val: function cap
:param b1:0 >= some value <= 1
:param num_samples: the number of values to generate
:return: function values
```

4.8.2.2 visualize_function()

```
def Visualization.visualize_function (
              values,
              name,
              color,
              ax = None )
```

```
plot a function
:param values: function values
:param name: name of the function
:param color: color of te plot
:param ax: the axis on which to plot
:return: plot axis
```

4.8.2.3 visualize_market()

```
def Visualization.visualize_market (
              smart_contract,
              customer_list,
              shop_list,
              ax_cus = None,
              ax_shop = None,
              ax_cp = None,
              ax_ca = None )
```

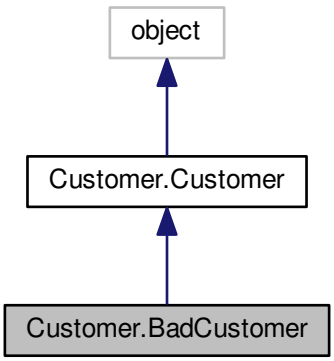
```
Function the visualizes the market
:param smart_contract: the smart contract containing the state of the block chain
:param customer_list: a list of customers in the block chain
:param shop_list: a list of shops in the block chain
:param ax_cus: the axis on which we plot the cumulative customer reputation
:param ax_shop: the axis on which we plot the cumulative shop reputation
:param ax_cp: the axis on which we plot the number of times a customer spent a coin
:param ax_ca: the axis on which cumulative shop coin count
:return: plot axes
```

Chapter 5

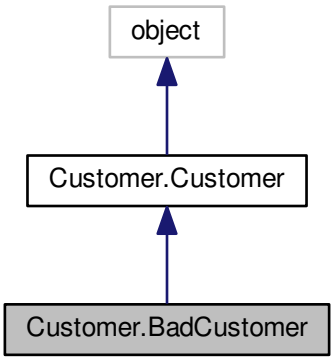
Class Documentation

5.1 Customer.BadCustomer Class Reference

Inheritance diagram for Customer.BadCustomer:



Collaboration diagram for Customer.BadCustomer:



Public Member Functions

- def **__init__** (self)
- def **choose_shop** (self, num_shops)

Public Attributes

- **recycle_prob**

Additional Inherited Members

5.1.1 Detailed Description

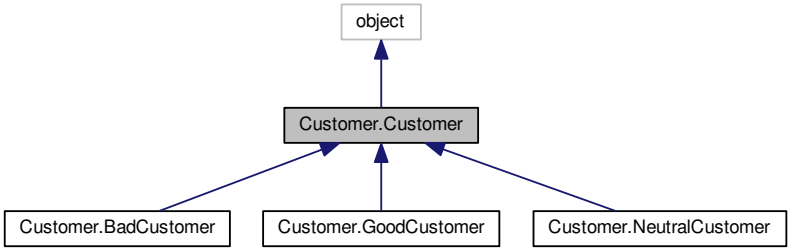
```
A bad customer recycles goods with a very low probability and always visits stores in an erratic
fashion
```

The documentation for this class was generated from the following file:

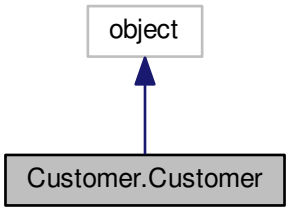
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Customer.py

5.2 Customer.Customer Class Reference

Inheritance diagram for Customer.Customer:



Collaboration diagram for Customer.Customer:



Public Member Functions

- def **__init__** (self, type_)
- def transfer_coin (self, coin_count)
- def set_coin (self, coins)
- def transfer_reputation (self, reputation, shop_address)
- def set_reputation (self, shop_address, reputation)
- def get_address (self)
- def choose_to_recycle (self)
- def choose_to_pay_by_coin (self)
- def get_coin_spend (self)
- def get_coin (self)
- def get_reputation (self, shop_address)
- def get_type (self)
- def choose_shop (self, num_shops)

Public Attributes

- **customer_id**
- **reputation**
- **coins**
- **recycle_prob**
- **preferred_shop**
- **type_**

Static Public Attributes

- int **CUSTOMER_ID** = 0

5.2.1 Detailed Description

```
This is a generic Customer class
```

5.2.2 Member Function Documentation

5.2.2.1 choose_shop()

```
def Customer.Customer.choose_shop (
              self,
              num_shops )
```

```
A choose shop strategy that depends on what type of a customer one is.
:param num_shops: The total number of shops available
:return: The chosen shop index
```

5.2.2.2 choose_to_pay_by_coin()

```
def Customer.Customer.choose_to_pay_by_coin (
              self )
```

```
Takes a decision on whether this customer buys something with the coins he has
at a given instance in time.
:return: boolean flag indicating the customer's decision to buy with ReusabiliTokens
```

5.2.2.3 choose_to_recycle()

```
def Customer.Customer.choose_to_recycle (
              self )
```

Takes a decision on whether this customer recycles at a given time instance
:return: boolean flag indicating the customer's decision to recycle

5.2.2.4 get_address()

```
def Customer.Customer.get_address (
              self )
```

Get the address of this customer
:return: customer address on the block chain

5.2.2.5 get_coin()

```
def Customer.Customer.get_coin (
              self )
```

Get the number of coins that this customer has.
:return: None

5.2.2.6 get_coin_spend()

```
def Customer.Customer.get_coin_spend (
              self )
```

Get the number of coins this customer spends every time he buys something with ReusabiliTokens
:return: number of coins

5.2.2.7 get_reputation()

```
def Customer.Customer.get_reputation (
              self,
              shop_address )
```

Get the reputation that this customer has earned from a specific store
:param shop_address: the shop address for which this customer's reputation is being queried
:return: None

5.2.2.8 get_type()

```
def Customer.Customer.get_type (
              self )
```

Get the type of the Customer. Currently implemented customers can either be Good, Bad or Neutral
:return: None

5.2.2.9 set_coin()

```
def Customer.Customer.set_coin (
              self,
              coins )
```

Initialize coins for this customer
:param coins: coin count
:return: None

5.2.2.10 set_reputation()

```
def Customer.Customer.set_reputation (
              self,
              shop_address,
              reputation )
```

Initialize reputation for this customer
:param shop_address: the shop for which the reputation is being initialized
:param reputation: the reputation to give to the customer
:return: None

5.2.2.11 transfer_coin()

```
def Customer.Customer.transfer_coin (
              self,
              coin_count )
```

Transfer coins to the customer for book keeping
:param coin_count: coin count
:return: None

5.2.2.12 transfer_reputation()

```
def Customer.Customer.transfer_reputation (
              self,
              reputation,
              shop_address )
```

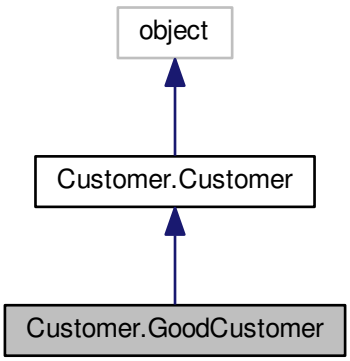
```
Transfer reputation to this customer from a specific shop
:param reputation: reputation coins
:param shop_address: the shop for which the reputation is being generated
:return: None
```

The documentation for this class was generated from the following file:

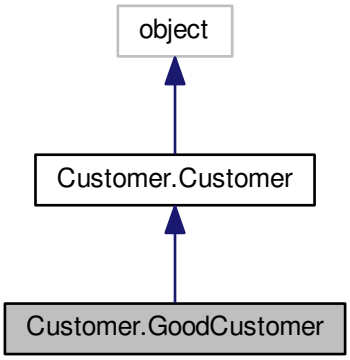
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Customer.py

5.3 Customer.GoodCustomer Class Reference

Inheritance diagram for Customer.GoodCustomer:



Collaboration diagram for Customer.GoodCustomer:



Public Member Functions

- def **__init__** (self)
- def **choose_shop** (self, num_shops)

Public Attributes

- **recycle_prob**
- **preferred_shop**

Additional Inherited Members

5.3.1 Detailed Description

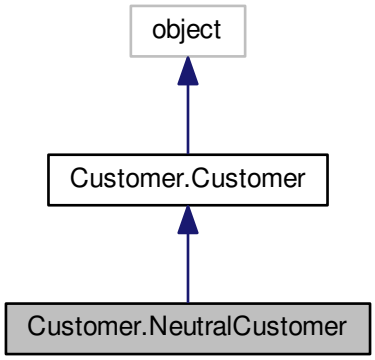
```
A good customer recycles goods with a high probability and always revisits his favorite store
to maximize his reputation.
```

The documentation for this class was generated from the following file:

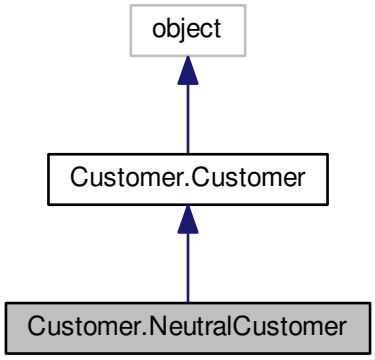
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Customer.py

5.4 Customer.NeutralCustomer Class Reference

Inheritance diagram for Customer.NeutralCustomer:



Collaboration diagram for Customer.NeutralCustomer:



Public Member Functions

- def **__init__** (self)
- def **choose_shop** (self, num_shops)

Public Attributes

- **recycle_prob**
- **preferred_shops**
- **preference_ratio**

Additional Inherited Members

5.4.1 Detailed Description

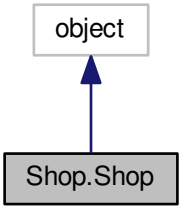
```
A neutral customer recycles his goods with a decent probability and revisits a small subset of all stores
that he likes to visit.
```

The documentation for this class was generated from the following file:

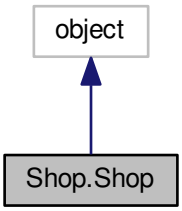
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Customer.py

5.5 Shop.Shop Class Reference

Inheritance diagram for Shop.Shop:



Collaboration diagram for Shop.Shop:



Public Member Functions

- def **__init__** (self)
- def get_shop_address (self)
- def get_coin_count (self)
- def buy_with_coins (self, coins)
- def pay_dues_to_smart_contract (self, smart_contract)

Public Attributes

- **shop_id**
- **name**
- **coin_count**

Static Public Attributes

- int **SHOP_ID** = 0

5.5.1 Detailed Description

```
This is an implementation of a generic shop that accepts ReusabiliTokens
```

5.5.2 Member Function Documentation

5.5.2.1 buy_with_coins()

```
def Shop.Shop.buy_with_coins (
              self,
              coins )
```

```
Every time a customer decides to buy with ReusabiliTokens from this shop, the shop
gets that many ReusabiliTokens
:param coins: number of coins that this shop gets
:return: None
```

5.5.2.2 get_coin_count()

```
def Shop.Shop.get_coin_count (
              self )
```

```
Get the number of coins that this shop is in possession of.
:return: coin count of the shop
```

5.5.2.3 get_shop_address()

```
def Shop.Shop.get_shop_address (
              self )
```

```
Get the address of this shop.
:return: shop address
```

5.5.2.4 pay_dues_to_smart_contract()

```
def Shop.Shop.pay_dues_to_smart_contract (
              self,
              smart_contract )
```

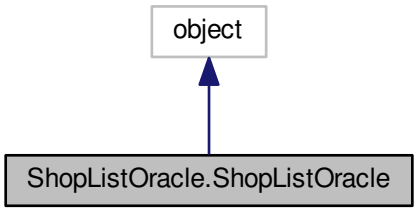
```
This function makes this shop pay whatever ReusabiliTokens it has to the smart contract.
:param smart_contract: the smart contract to which the payment should be made
:return: None
```

The documentation for this class was generated from the following file:

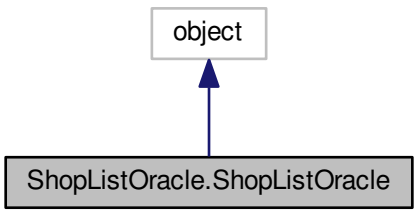
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Shop.py

5.6 ShopListOracle.ShopListOracle Class Reference

Inheritance diagram for ShopListOracle.ShopListOracle:



Collaboration diagram for ShopListOracle.ShopListOracle:



Public Member Functions

- def **__init__** (self)
- def verify_shop (self, shop_address)
- def register_new_shop (self, shop_address)

Public Attributes

- **shop_list**

5.6.1 Detailed Description

```
This is an implementation of an Oracle that checks if a specific address belongs
to a registered shop.
```

5.6.2 Member Function Documentation

5.6.2.1 register_new_shop()

```
def ShopListOracle.ShopListOracle.register_new_shop (
              self,
              shop_address )
```

```
This function is used to register a new shop with the oracle.
:param shop_address: the address of the shop to register
:return: None
```

5.6.2.2 verify_shop()

```
def ShopListOracle.ShopListOracle.verify_shop (
              self,
              shop_address )
```

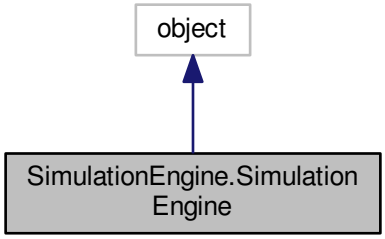
```
The smart contract calls this function to verify if the given address is indeed a shop address
:param shop_address: the address to verify
:return: returns True if the given address is indeed that of a registered shop
```

The documentation for this class was generated from the following file:

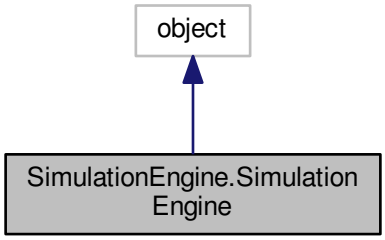
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/ShopList↩ Oracle.py

5.7 SimulationEngine.SimulationEngine Class Reference

Inheritance diagram for SimulationEngine.SimulationEngine:



Collaboration diagram for SimulationEngine.SimulationEngine:



Public Member Functions

- def __init__ (self, num_customers, num_shops, sim_iters, coin_limit, rep_limit, coin_rep_factor, payment_↩ due)
- def run (self, claim_failure_probability=0.00001)

Public Attributes

- **num_customers**
- **num_shops**
- **sim_iters**
- **customers**
- **shops**
- **time_oracle**
- **shop_list_oracle**

- **address**
- **smart_contract**
- **coin_limit**
- **rep_limit**
- **payment_due**
- **coin_rep_factor**

5.7.1 Detailed Description

```
This is the engine that runs the market simulation
```

5.7.2 Constructor & Destructor Documentation

5.7.2.1 __init__()

```
def SimulationEngine.SimulationEngine.__init__ (
              self,
              num_customers,
              num_shops,
              sim_iters,
              coin_limit,
              rep_limit,
              coin_rep_factor,
              payment_due )

Constructor
:param num_customers: the number of customers in the market
:param num_shops: the number of shops in the market
:param sim_iters: the number of iterations to simulate the market
:param coin_limit: the maximum coin that a customer can possess (NOT USED CURRENTLY)
:param rep_limit: the maximum reputation that a customer can earn
:param coin_rep_factor: a multiplicative factor to convert reputations to ReusabiliTokens
:param payment_due: the duration over which shops have to pay back their dues
```

5.7.3 Member Function Documentation

5.7.3.1 run()

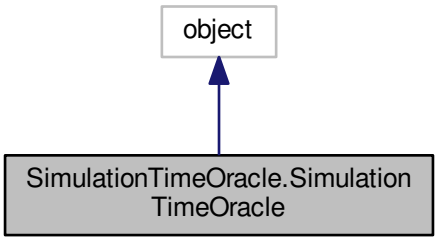
```
def SimulationEngine.SimulationEngine.run (
              self,
              claim_failure_probability = 0.00001 )

Run the simulator
:param claim_failure_probability: probability of a customer claim being false
:return: None
```

The documentation for this class was generated from the following file:

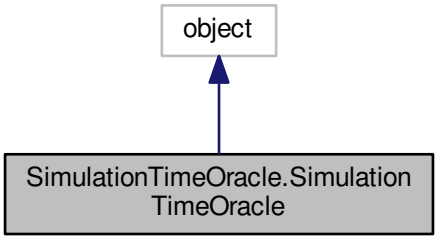
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Simulation↩ Engine.py

5.8 SimulationTimeOracle.SimulationTimeOracle Class Reference

Inheritance diagram for SimulationTimeOracle.SimulationTimeOracle:



Collaboration diagram for SimulationTimeOracle.SimulationTimeOracle:



Public Member Functions

- def **__init__** (self)
- def increment_time (self)
- def get_time (self)

Public Attributes

- **time**

5.8.1 Detailed Description

```
An oracle that gives the True time
```

5.8.2 Member Function Documentation

5.8.2.1 get_time()

```
def SimulationTimeOracle.SimulationTimeOracle.get_time (
              self )
```

```
A smart contract would call this function on the oracle to determine some standard wall clock
time
:return: False
```

5.8.2.2 increment_time()

```
def SimulationTimeOracle.SimulationTimeOracle.increment_time (
              self )
```

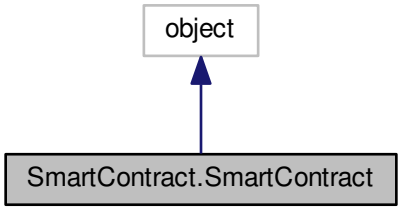
```
Increment the wall clock time on the Oracle
:return:
```

The documentation for this class was generated from the following file:

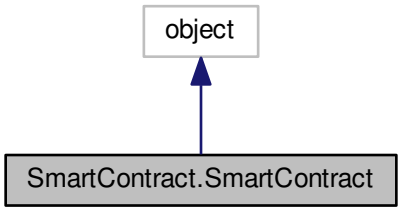
- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Simulation↩ TimeOracle.py

5.9 SmartContract.SmartContract Class Reference

Inheritance diagram for SmartContract.SmartContract:



Collaboration diagram for SmartContract.SmartContract:



Public Member Functions

- def **__init__** (self, owner_address)
- def **set_oracle** (self, sender_address, shop_oracle, time_oracle)
- def **set_coin_limit** (self, sender_address, coin_limit)
- def **set_reputation_limit** (self, sender_address, reputation_limit)
- def **set_coins_per_reputation_token** (self, sender_address, factor)
- def **set_payment_duration** (self, sender_address, duration)
- def **check_payments** (self, sender_address, current_time)
- def **make_payment** (self, shop_address, payment)
- def **make_claim** (self, shop_address, customer_address)
- def **verify_claim** (self, shop_address, customer_address)
- def **customer_buys_with_coin** (self, customer_address, shop_address, num_coins)
- def **deteriorate_customer_reputation** (self, sender_address, value=0.05)
- def **calculate_shop_reputation** (self, shop_address)
- def **calculate_customer_reputation** (self, customer_address)
- def **valid_shops_left** (self, shop_addresses)
- def **get_coin_map** (self)
- def **get_reputation_map** (self)
- def **get_coin_purchase_map** (self)

Public Attributes

- **coin_map**
- **reputation_map**
- **customer_recycle_map**
- **shop_coin_map**
- **cus_buys_with_coin_map**
- **status**
- **current_customer_address**
- **current_shop_address**
- **owner_address**
- **shop_oracle**
- **time_oracle**
- **known_shops**
- **black_listed_shops**

- **shop_payment_times**
- **reputation_limit**
- **coin_limit**
- **coins_per_reputation_token**
- **payment_due_date**

The documentation for this class was generated from the following file:

- /home/prash/workspace/dev_space/demo_apps/ReusabiliToken/ReusabiliTokenSimulator/src/Smart↩ Contract.py