

The Bank Simulation

Object Oriented Programming With Java



Umut Akkaya 150315031

DEVRİM NAZ AKTAŞ 150315042

# Index

1. [Purpose](#_Purpose)
2. [Document](#_Document)
   1. [**Collections**](#_Collections)
      1. [Bank Collection](#_Bank_Collections)
      2. [Customer Collection](#_Customer_Collections)
      3. [History Collection](#_History_Collections)
   2. [**Bank**](#_Bank)
   3. [Person](#_Person)
   4. [Customer](#_Customer)
   5. [**History**](#_History)
   6. [Event](#_Event)
   7. [Sex](#_Sex)
3. [**Scheme**](#_Scheme)

# Purpose

You are going to implement a "bank simulation" in Java. The bank has customers and each customer should have at least these properties:

* Customer ID : The unique number that separates one customer from the other.
* (Beware that two customers with the same ID cannot be added)
* Name : Customer’s name
* Last name : Customer’s last name
* Sex(male/female) : Customer's gender
* Credit : The amount of money that customer has

Your simulation must supply at least these operations:

* Adding a new customer
* Deleting an existing customer (by customer ID)
* Listing all customers
* Listing customers by customer ID, name, last name or gender
* Updating a customers information (customer name, last name, gender)
* Adding money to bank, i.e raising credit
* Drawing money from the bank, i.e reducing credit (beware that credit cannot be negative !)

Please follow the rules of good programming ! (try to implement loose coupling and a high cohesive system, avoid code duplication, use object oriented programming encapsulation techniques, etc.)

# Document

## Collections

### Bank Collections

This is used for collecting banks

#### Methods

**Constractors**

**BankCollection**()

This is create a new unlimited bank list

**BankCollection**(long **capacity**)

*capacity is limit of bank list, and type is Long*

This is create a new limited bank list

boolean **add**(Bank **bank**)

*bank variable new type of Bank*

This is add new bank in banklist, and return true, or false. İf banklist is full return false otherwise return true

boolean **remove**(long **id**)

*id is bank order number, and type is Long*

This is remove bank in banklist, and return true, or false. İf bank is removed return true otherwise return false

Bank **get**(long **id**)

id is bank order number, and type is Long

This is get bank in banklist, and return type of Bank.

Map<Long, Bank> **searchBanksWithName**(String **name**)

name is name of the bank, and type is String

This is get list of bank by start with bank name equal to name. Return type is Map<Long,Bank>

Map<Long, Bank> **getAll**()

This is get all bank in banklist. Return type is Map<Long,Bank>

long **getSize**()

This is return a count of bank

long **getLastId**()

This is return a last registered bank id

### Customer Collections

This is used for collecting customers

#### Methods

**Constractors**

**CustomerCollection**()

This is create an unlimited customer collections

**CustomerCollection**(long **capacity**)

*capacity is limit of the customerlist, and type is Long*

This is create a limited customer collection

boolean **add**(Customer **customer**)

*customer is new type of Customer*

This is add a new customer in customer collection, and return true if customer collection has a enough capacity.

boolean **remove**(long **id**)

*id is customer id, and type of Long*

This is remove customer by id of customer, and return true if customer is removed

Customer **get**(long **id**)

*id is customer id, and type of Long*

This is get customer by id of customer,and return type of Customer

Map<Long, Customer> **searchWithFullName**(String **name**)

*name is name of customer, and type of String*

This is get all customer which are contains name in these names, and type is Map<Long, Customer>

Map<Long, Customer> **getAll**()

This is return all customer in customer list, and return type is Map<Long, Customer>

long **getSize**()

This is return count of customer,and return type is Long

long **getLastId**()

This is return last id of new customer, and type is Long

### History Collections

This is collecting a history of bank and user

#### Methods

**Constractor**

**HistoryCollection**()

This is new instance of HistoryCollection

boolean **containsKey**(long **id**)

*id is order number of history,and type is Long*

This is check history collection has a id of history, and return true if list contain id

void **add**(Customer **customer**, Event **event**, String **info**)

*customer which is doing event.*

*event is what happend*

*info is information of event*

This is add a new event in historycollection

History **get**(long **id**)

*İd is id of history, and type is Long*

This is get history which id’s equal id,and return type of History

Map<Long, History> **getAll**()

This is return all history of bank, and return Map<Long, History>

Map<Long, History> **getHistoryOfUser**(long **customerId**)

*customerId is id of customer, and type is Long*

This is return histories of user, and return type of Map<Long, History>

## Bank

New bank instance has a customer and collect customer informations which are amount of money customer name customer lastname customer birthday, and others.

#### Methods

**Constractors**

**Bank**(String **name**)

*name is name of bank, type is String*

This creates new instance of bank and give a name

**Bank**(String **name**, long **capacity**)

*name is name of bank, type is String*

*capacity is capacity of customers in bank, and type is Long*

This creates new instance of bank, and give a name and capacity of bank

boolean **addCustomer**(Person **person**, double **balance**)

*person is customer information, and type is Person*

*balance is starting balance*

This is create new customer and give starting balance,return true if customer is added

boolean **removeCustomer**(long **id**)

*id is id of customer, and type is Long*

This is remove customer by id in bank return true if customer is removed

Customer **getCustomer**(long **id**)

*İd is id of customer, and type is Long*

This is get customer by id, and return type of Customer

Map<Long, Customer> **getAllCustomers**()

This is return all cutomer in bank, return type is Map<Long, Customer>

Map<Long, Customer> **searchCustomersWithFullName**(String **name**)

*name is name of customer*

This is return customers which are contains name parameter in full name, and return Map<Long, Customer>

long **getCustomerCount**()

This return customer count in bank, and return type is Long

long **getLastCustomerId**()

This is return last customer id in bank, and return type is Long

History **getHistory**(long **id**)

*id is id of history, and type of Long*

This return history by id, and return type of History

Map<Long, History> **getAllHistory**()

This return all history of bank, return type is Map<Long, History>

Map<Long, History> **getHistoryOfUser**(long **customerId**)

*customerId is id of customer, and type of Long*

This retrun history of user, and return type is Map<Long, History>

String **getName**()

This is return name of bank, and type is String

boolean **setName**(String **name**)

*name is name of bank, and type of String*

This is set name of bank

## Person

This is new instance of person.This include a personal informations birthday, name, lastname, etc.

#### Methods

**Constractors**

**Person**(String **name**, String **lastName**, Sex **sex**, Date **birthDay**)

*name is first name of person, and type is String*

*lastName is last name of person, and type is String*

*sex is gender of person, and type is Sex*

*birthDay is birthday of user, and type is Date*

This is create new instance of person

**Person**(String **name**, String **lastName**)

*name is first name of person, and type is String*

*lastName is last name of person, and type is String*

This is create new instance of person

String **getFullName**()

This is combine first name and last name of person and return it, return type is String

String **getName**()

This is return first name of person, return type is String

boolean **setName**(String **name**)

*name is person name, type of String*

This is set a name of person return true if name is changed

String **getLastName**()

This is get last name of person, return type is String

boolean **setLastName**(String **lastName**)

*lastName is last name of person, type of String*

This is set a last name of person return true if last name is changed

Sex **getSex**()

This is get sex of person, return type is Sex

boolean **setSex**(Sex **sex**)

*sex is person sex, type of Sex*

This is set a sex of person return true if sex name is changed

Date **getBirthDay**()

This is return birthday of person, return type is Date

boolean **setBirthDay**(Date **birthDay**)

*birthDay is birthday of person, type is Date*

This is set birth day of person, return true if birthdat is changed

## Customer

This is new instance of customer. This is extends from person, and this include money amount, and id of customer.

#### Methods

**Constractors**

**Customer**(Person **person**)

*person is personanl information, and type is Person*

This is create new instance of customer by person

**Customer**(Person **person**, double **balance**)

*person is personanl information, and type is Person*

*balance is starting balance*

This is create new instance of customer by person and balance

**Customer**(String **name**, String **lastName**, Sex **sex**, Date **birthDay**, double **balance**)

*name is first name of person, and type is String*

*lastName is last name of person, and type is String*

*sex is gender of person, and type is Sex*

*birthDay is birthday of user, and type is Date*

*balance is starting balance*

This is create new instance of customer by name, last name, sex, birthday, balance

**Customer**(String **name**, String **lastName**)

*name is first name of person, and type is String*

*lastName is last name of person, and type is String*

This is create new instance of customer by name, and last name

long **getId**()

This is return id of customer, return type is Long

double **getBalance**()

This is return balance of customer, return type is Double

boolean **drawMoney**(double **amount**)

*amount is money amount which is money of person*

This is draw money, and return true if money enough

boolean **depositMoney**(double **amount**)

*amount is money amount which is money of person*

This is deposit money, and return true if money enough

## History

This is history. It is collect event,info, and customer

#### Methods

**Constractor**

**History**(Customer **customer**, Event **event**, String **info**)

*customer which is done something for save history, type is Customer*

*event is what happend customer,type is Event*

*info is information of event*

This is create instance of history

Date **getTime**()

This is return time of history, return type is Date

Customer **getCustomer**()

This is retun customer which done event,return type is Customer

Event **getEvent**()

This is return event, return type is Event

String **getInfo**()

This is return info of event, return type is String

## Event

This is event enum, and has five event type

#### Enums

**RemoveCustomer**

**NewCustomer**

**Deposit**

**Draw**

**UpdateProfil**

## Sex

This is gender enum, and has three gender type

#### Enums

**Undefined**

**Male**

**Female**

# Scheme

