תורה 1

תרגול 5: מחלקי עצמים (חקל ב')
ליאור שפריר ואוהד ברזילי
Class Diagram

BankAccount

- public BankAccount(Customer customer, long id)
- public void withdraw(double amount)
- public void deposit(double amount)
- public void transferTo(double amount, BankAccount otherAccount)
- public double getBalance()
- public Customer getOwner()
- public double getNumber()

Customer

- public Customer(String name, String id)
- public String getName()
- public String getID()

Bank

main

...
public class Customer {
    public Customer(String name, String id) {
        this.name = name;
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public String getID() {
        return id;
    }
    private String name;
    private String id;
}

The Customer Class
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Customer:
String name:
String id:

Bank
main (...) {
   customer1:
}

String:
"Avi Cohen"

String:
"025285244"
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);
        System.out.println("account1 has "+ account1.getBalance());
        System.out.println("account2 has "+ account2.getBalance());
    }
}
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);
        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Object Diagram

BankAccount:
- owner:
- balance: 0
- number: 1234

BankAccount:
- owner:
- balance: 0
- number: 5678

Bank
- main (...):
  - customer1:
  - customer2:
    - account1:
    - account2:

Customer:
- String name:
- String id:
  - String: "Avi Cohen"
  - String: "025285244"

Customer:
- String name:
- String id:
  - String: "Rita Stein"
  - String: "024847638"
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer1, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Bank

main (...) {
    customer1:
    customer2:
    account1:
    account2:
    account3:
}

Customer:
String name:
String id:

Customer:
String name:
String id:

Customer:
String name:
String id:

String: "Avi Cohen"

String: "025285244"

String: "Rita Stein"

String: "024847638"
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Message Sequence Chart

main

account1

owner:
balance: 0
number: 1234

1000

deposit

account2

owner:
balance: 0
number: 5678

account3

owner:
balance: 0
number: 2984

מיהו拍摄 במהל
 выполнен deposit ?deposit ביצוע

this the possessor
Message Sequence Chart

- main
  - account1
    - owner: balance: 1000 number: 1234
    - deposit: 1000
  - account2
    - owner: balance: 0 number: 5678
  - account3
    - owner: balance: 0 number: 2984
  - void
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Message Sequence Chart

main

account1

owner:
balance: 1000
number: 1234

deposit

account2

owner:
balance: 500
number: 5678

deposit

account3

owner:
balance: 0
number: 2984

deposit

void

1000

500

מיהו המהלך this?
deposit ביצועים

time
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Message Sequence Chart

main -> account1
- owner: balance: 1000 number: 1234
- deposit 1000
- void

account1 -> account2
- deposit 500
- void

account2 -> account3
- deposit

main
- transferTo 100, account3

account3
- owner: balance: 0 number: 2984

This mühow bemahal? transferTo bizevut?
Message Sequence Chart

main

account1
owner: balance: 900
number: 1234

account2
owner: balance: 500
number: 5678

account3
owner: balance: 100
number: 2984

1000 deposit
void

500 deposit
void

100, account3 transferTo
void

100 deposit
void

מייהו במעלון؟
deposit

this ביצוע

במהל

ביצוע deposit
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
Message Sequence Chart

[Diagram showing interactions between different accounts and transactions such as deposit, withdraw, transferTo, and void.]

Example transactions:
- **main** deposits 1000 to **account1**
- **account1** has owner: balance: 900, number: 1234
- **main** deposits 500 to **account2**
- **account2** has owner: balance: 200, number: 5678
- **account1** transfers 100 to **account3**
- **account3** has owner: balance: 100, number: 2984
- **account1** withdraws 300
- **account2** deposits 100

Note: For a full understanding, consult the diagram and read the text carefully.
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer2, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);
        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}