

Composition

Usa Sammapun

Composition

- Has an instance variable with an object type
 - Has-A relationship between class/objects
- Instance variables can be divided into two categories
 - primitive type (and String type) attribute, fields
 - object type composition

Composition

- Composition helps break long code into
 - several pieces of modular code
- Benefits of composition (modular code)
 - easy to understand
 - easy to reuse code
 - easy to change / extend feature
 - easy to debug

https://github.com/ladyusa/atm

```
Customer
id name pin checkPin()

has-a id name balance deposit() withdraw()
```

```
public class BankAccount {
   private int id;
   private String name
   private double balance;
   // . . . code . . .
public class Customer {
   private int id;
                               attribute, field
   private String name
   private int pin;
   private BankAccount account; } composition
   // . . . code . . .
```

DataSourceDB

readCustomers()

```
public class DataSourceDB {
    public Map<Integer, Customer> readCustomers() {
        // . . . code . . .
    }
}
```

```
public class Bank {
   private String name
   private Map<Integer,Customer> customers;
   private DataSourceDB dataSource;
   public Bank(String name) {
      this.name = name;
      this.dataSource = new DataSourceDB();
      this.customers = dataSource.readCustomers();
   }
                                                               has-a
                                          has-a
                                                                       BankAccount
                                                   Customer
   // . . . code . . .
                               Bank
                                                 id
                                          has *
                                                               has
                         name
                                                                       name
                                                 name
}
                         registerCustomer()
                                                                       balance
                                                 pin
                         findCustomer()
                                                                       deposit()
                                                 checkPin()
                                                                       withdraw()
                           DataSourceDB
                          readCustomers()
```

```
public class ATM {
   private Bank bank;
   private Customer loginCustomer;
   public ATM() {
       this.bank = new Bank();
       this.loginCustomer = null;
   }
   // . . . code . . .
        ATM
                                                                          BankAccount
                                                     Customer
  validateCustomer()
                                                                          id
                                 Bank
                                                   id
  deposit()
                           name
                                                                          name
  withdraw()
                                                   name
                           registerCustomer()
                                                                          balance
                                                   pin
  getBalance()
                           findCustomer()
                                                                          deposit()
                                                   checkPin()
  transfer()
                                                                          withdraw()
  end()
                            DataSourceDB
                           readCustomers()
```

```
public class AtmUI {
      private ATM atm;
      public AtmUI() {
          this.atm = new ATM();
      }
      // . . . code . . .
                     ATM
                                                                                       BankAccount
                                                                  Customer
              validateCustomer()
                                                                                       id
                                             Bank
                                                                id
              deposit()
AtmUI
                                                                                       name
                                       name
                                                                name
              withdraw()
                                       registerCustomer()
run()
                                                                                       balance
              getBalance()
                                                                pin
                                       findCustomer()
                                                                                       deposit()
                                                                checkPin()
              transfer()
                                                                                       withdraw()
              end()
                                         DataSourceDB
                                        readCustomers()
```

```
public class Main {
    public static void main(String[] args) {
        AtmUI atmUI = new AtmUI();
        atmUI.run();
    }
}
```