

## **Problem 2: Order food in a food delivery app**

### **Objects and behaviors:**

UberEats:

Data:

Behavior: login, showMenu, callRestaurant, callDriver

Restuarant:

Data: name

Behavior: doneOrder

User:

Data: userName

Behavior: takeOrder, checkOut, rate

CreditCard:

Data:

Behavior: authorizeTransaction

Driver:

Data: driverName

Behavior: diliver

### **Sequence of invoking behaviors on objects**

User Ash -> new User(username, password)

UberEats.login -> username, password

If(username, password -> correct)

UberEats.showMenu

Else

Return null

Ash.takeOrder -> courseName, address, phoneNum, cardInfo ->

Ash.checkout -> creditCard.authorizeTransaction

UberEats.callRestuarant

Restaurant.doneOrder

UberEats.callDriver -> checkAvailableDrivers

Loop

If (quantity of driver >= 1)

Driver.deliver

Else

UberEats.addMoneyForDriver

Ash.rate

### **Problem 3: Design a platform for buying tickets of local event**

#### **Objects and behaviors:**

Platform:

Data: Name

Behavior: login, register, showEvents, callEventHandler

Event:

Data: name

Behavior: checkCapacity, processTicket

User:

Data: userName

Behavior: selectEvent, takeOrder, checkOut

CreditCard:

Data:

Behavior: authorizeTransaction

Email:

Data:

Behavior: sendTicket

### **Sequence of invoking behaviors on objects**

User Ash -> new User(username, password)

Platform.login -> username, password

If(username, password -> correct)

Platform.showEvents

Else

Return Platform.register

Ash.takeOrder -> eventName, numberOfGroup, email, cardInfo ->

Ash.checkout -> creditCard.authorizeTransaction

Platform.callEventHolder

If(Event.checkCapacity -> tickets >= numberOfGroup)

Event.processTicket

Email.sentTicket

Else

failToOrder

#### **Problem 4: Buy a computer from Amazon**

##### **Objects and behaviors:**

Amazon:

Data: Name

Behavior: login, register, showTraders, checkCapacity.processOrder

Trader:

Data: name

Behavior:

User:

Data: userName

Behavior: search, takeOrder

CreditCard:

Data:

Behavior: authorizeTransaction

### **Sequence of invoking behaviors on objects**

User Ash -> new User(username, password)

Amazon.login -> username, password

If(username, password -> correct)

login

Else

Return Platform.register

Ash.search -> computer

Amazon.showTraders

Ash.takeOrder -> itemName, numberOfItem, address, phoneNumber, cardInfo ->

Loop

Amazon.checkCapacity

If(amount >= numberOfItem)

Success

Ash.checkout -> creditCard.authorizeTransaction

If (success)

Amazon.processOrder

Else

failToOrder

Else

Fail to order

## **Problem 5: Design an app for booking hotels**

### **Objects and behaviors:**

HotelPlatform:

Data: Name

Behavior: login, register, showHotels, checkCapacity

Hotel:

Data: name

Behavior: processTicket

User:

Data: userName

Behavior: search, takeOrder, rate

CreditCard:

Data:

Behavior: authorizeTransaction

Email:

Data: hotelEmail

Behavior: sendTicket

### **Sequence of invoking behaviors on objects**

User Ash -> new User(username, password)

hotelPlatform.login -> username, password

If(username, password -> correct)

Platform.showHotels

Else

Return Platform.register

Ash.search -> addressOfHotels ->

hotelPlatform.showHotels

Ash.takeOrder -> email, numbersOfNight, cardInfo ->

hotelPlatform.checkCapacity

Loop

If(capacity >= numbersOfNight)

Hotel.processTicket

Email.sendTicket

Else

failToOrder

Ash.rate