

1. Class “InterbankInterface”

<<interface>> InterbankInterface	
+ <<exception>> <i>payOrder(card : CreditCard, amount : int, contents : String) : PaymentTransaction</i> + <<exception>> <i>refund(card : CreditCard, amount : int, contents : String) : PaymentTransaction</i> + <<exception>> <i>getBalance(card : CreditCard) : int</i>	

Attribute

None

Operation

No	Name	Return type	Description (purpose)
1	payOrder	PaymentTransaction	Pay order, and then return the payment transaction
2	refund	PaymentTransaction	Refund, and then return the payment transaction
3	getBalance	int	Get the balance and then return it

Parameter:

- card – the credit card used for payment/refund
- amount –the amount to pay/refund
- contents – the transaction contents

Exception:

- PaymentException – if responded with a pre-defined error code
- UnrecognizedException – if responded with an unknown error code or something goes wrong

Method:

None

State

None

2. Class “PaymentController”

<<control>> PaymentController	
- card : CreditCard - interbank : InterbankInterface	
+ payOrder(amount : int, contents : String, cardNumber : String, cardHolderName : String, expirationDate : String, securityCode : String) : Map<String,String> - getExpirationDate(date : String) : String	

Attribute

No	Name	Data type	Default value	Description
1	card	CreditCard	NULL	Represent the card used for payment
2	interbank	InterbankInterface	NULL	Represent the Interbank subsystem

Operation

No	Name	Return type	Description (purpose)
1	payOrder	Map<String,String>	Pay order, and then return the result with a message

Parameter:

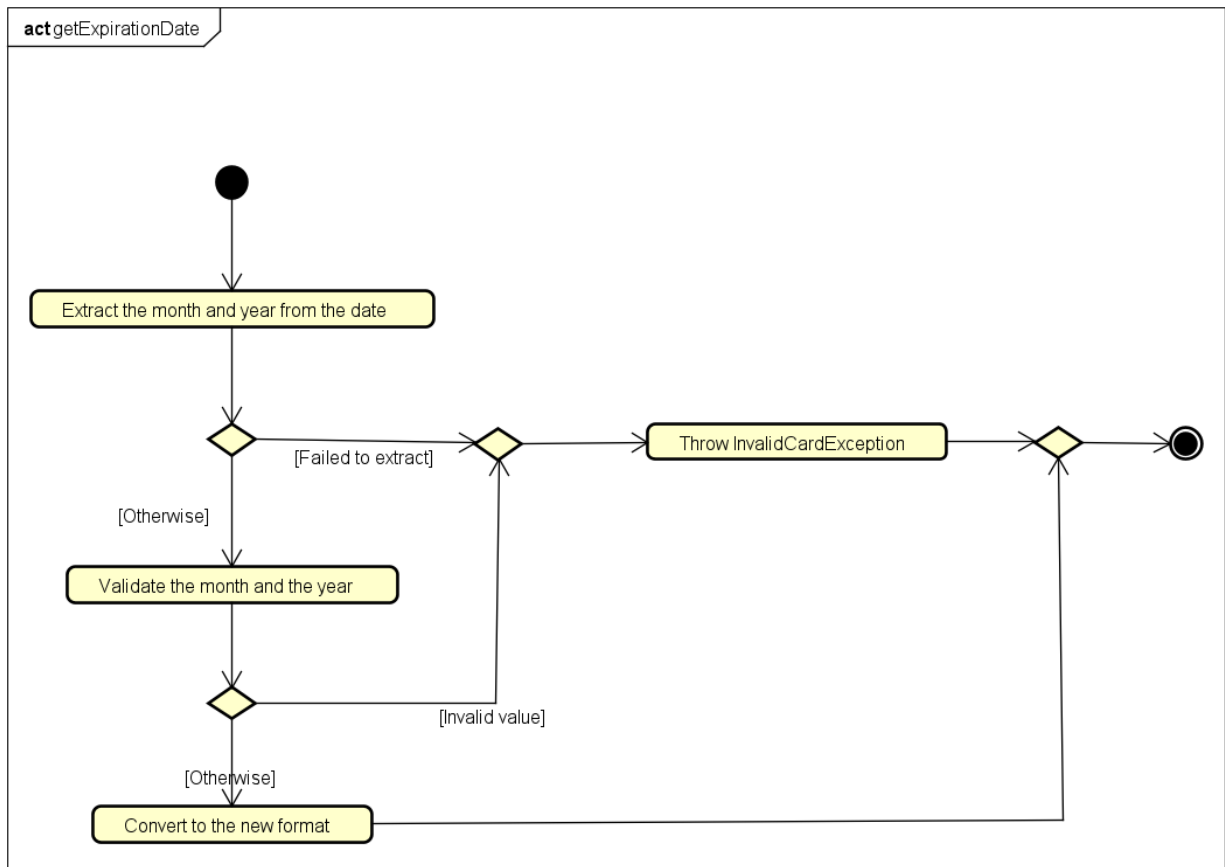
- amount – the amount to pay
- contents – the transaction contents
- cardNumber – the card number
- cardHolderName – the card holder name
- expirationDate – the expiration date in the format “mm/yy”
- securityCode – the cvv/cvc of the credit card

Exception:

- None

Method:

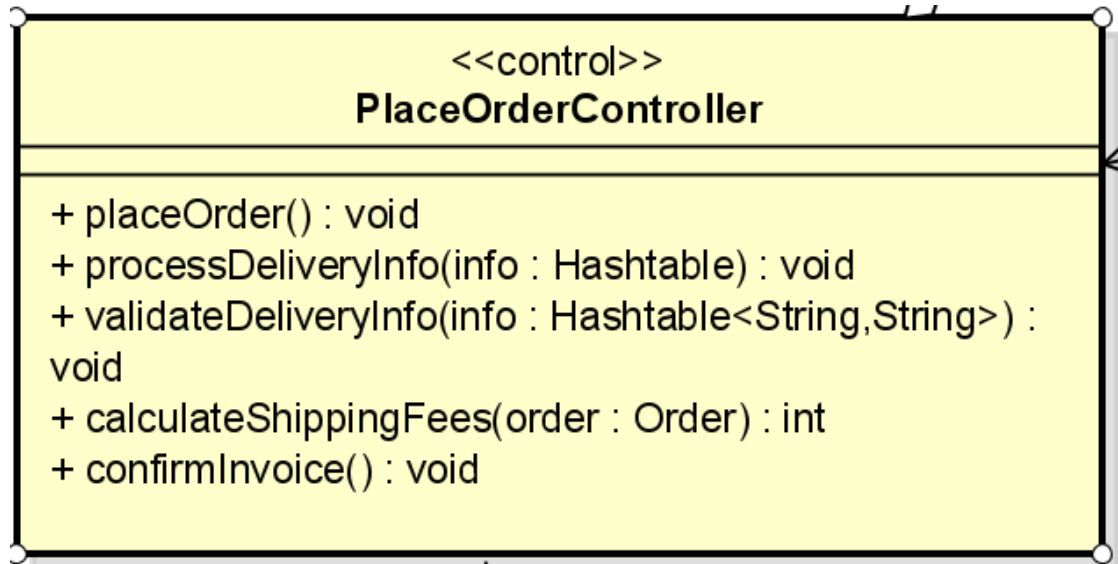
- getExpirationDate: Given the String “date” representing the expiration date in the format “mm/yy”, this method convert it into the required format “mmyy”. The algorithm is illustrated as follows.



State

None

3. Place order controller



Attribute

None

Operation

No	Name	Return type	Description (purpose)
1	placeOrder	void	Place a new order
2	processDeliveryInfo	void	Process the delivery information
3	validateDeliveryInfo	void	Validate the delivery information
4	calculateShippingFees	Int	Calculate the shipping fees by the formula
5	confirmInvoice	void	Confirm the information of Invoice

Parameter

- Info : the information of delivery
- order : the information of order

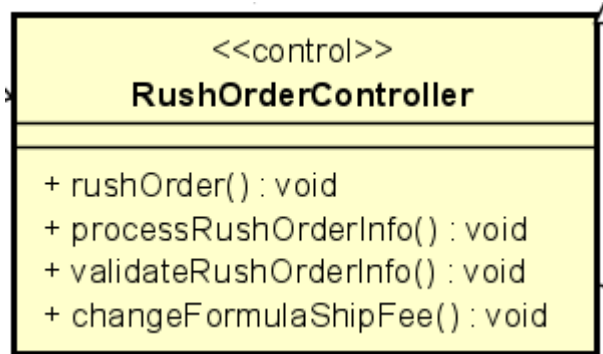
Exception

None

Method

None

4. Rush order controller



Attribute

None

Operation

No	Name	Return type	Description (purpose)
1	rushOrder	void	Place rush order , check if media and address are supported
2	processRushOrderInfo	void	Process the rush order information
3	validateRushOrderInfo	void	Validate the rush order information
4	changeFormulaShipFee	void	Change the formular of shipping fees

Parameter

None

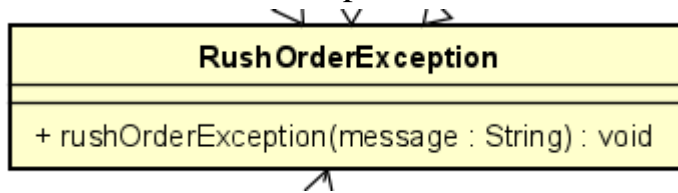
Exception

None

Method

None

5. Class “RushOrderException”



Attribute

None

Operation

No	Name	Return type	Description (purpose)
1	rushOrderException	void	Exceptions of rush order

Parameter

- message : the different message of exception

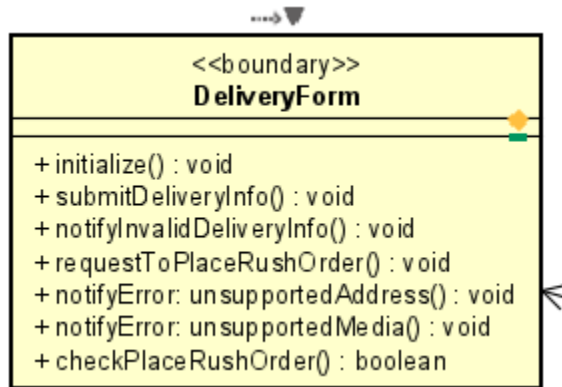
Exception

None

Method

None

6. Class “DeliveryForm”



Attribute

None

Operation

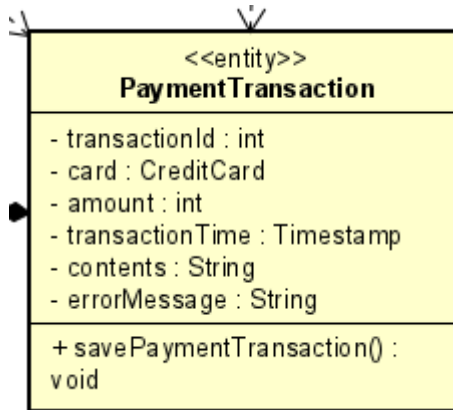
No	Name	Return type	Description (purpose)
1	initialize	void	Inititalize the delivery form
2	submitDeliveryInfo	void	Submit the delivery information
3	notifyInvalidDeliveryInfo	void	Notify that the delivery information is invalid
4	requestToPlaceRushOrder	void	Request to place rush order
5	notifyError: unsupportedAddress	void	Notify that the address is unsupported for rush order
6	notifyError: unsupportedMedia	void	Notify that the media is unsupported for rush order
7	checkPlaceRushOrder	Boolean	Check if user choose placing rush order

Parameter

Exception

Method

7. Class “PaymentTransaction”



Attribute

No	Name	Data type	Default value	Description
1	card	CreditCard	NULL	Represent the card used for payment
2	transactionId	int	NULL	Represent the id of transaction
3	amount	int	NULL	The amount to pay
4	transactionTime	Timestamp	NULL	Time of the transaction
5	contents	String	NULL	The transaction contents
6	errorMessage	String	NULL	The error message

Operation

No	Name	Return type	Description (purpose)
1	savePaymentTransaction	void	Save the information of payment transaction

Parameter

Exception

Method