1. Class “Product”

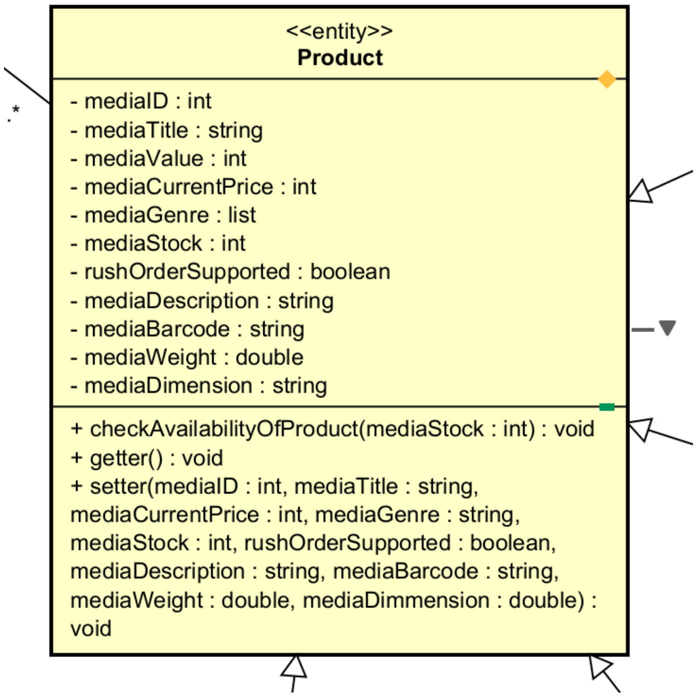

*
<<entity>>
Product
- mediaID : int
- mediaTitle : string
- mediaValue : int
- mediaCurrentPrice : int
- mediaGenre : list
- mediaStock : int
- rushOrderSupported : boolean
- mediaDescription : string
- mediaBarcode : string
- mediaWeight : double
- mediaDimension : string
+ checkAvailabilityOfProduct(mediaStock : int) : void
+ getter() : void
+ setter(mediaID : int, mediaTitle : string, mediaCurrentPrice : int, mediaGenre : string, mediaStock : int, rushOrderSupported : boolean, mediaDescription : string, mediaBarcode : string, mediaWeight : double, mediaDimmension : double) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	mediaID	int	N/A	Unique identifier for the product
2	mediaTitle	string	N/A	The name of the product
3	mediaValue			
4	mediaCurrentPrice	int	N/A	The current price of product (VND)
5	mediaGenre	list<String>	N/A	The list of product’s genres
6	mediaStock	int	N/A	The quantity of a particular product available in stock
7	rushOrderSupported	boolean	N/A	Indicates whether the product supports rush orders (true/false)
8	mediaDescription	string	N/A	Detailed description of the product
9	mediaBarcode	string	N/A	Barcode number associated with the product
10	mediaWeight	string	N/A	Weight of the product

11	mediaDimmension	string	N/A	Dimensions of the product

Table 2. Example of operation design

#	Name	Return type	Description (purpose)
1	checkAvailabilityOfProduct()	void	Used to check if a product is available in stock
2	getter()	v.v	Used to retrieve the values of the attributes in the Product class
3	setter(...)	void	Used to set the values for the attributes in the Product class

1. **Method "CheckAvailabilityOfProduct"**

- **Parameter**

#	Name	Description
1	mediaStock: int	The quantity of a particular product available in stock

- **Exception**

#	Name	Description
1	OutOfStockException	If the stock value is 0 or less

- **How to use parameter/attribute:** *checkAvailabilityOfProduct()* uses the parameter to check if the inventory stock is sufficient for the customer's order.

2. Class “DVD”

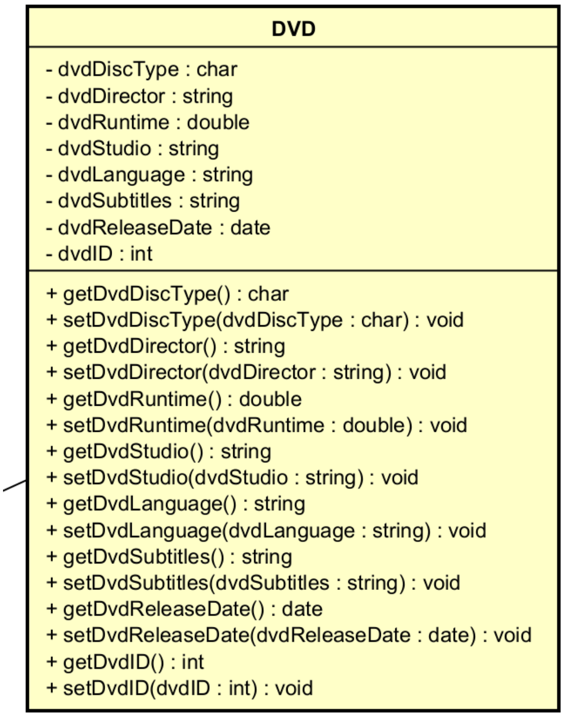

DVD
- dvdDiscType : char
- dvdDirector : string
- dvdRuntime : double
- dvdStudio : string
- dvdLanguage : string
- dvdSubtitles : string
- dvdReleaseDate : date
- dvdID : int
+ getDvdDiscType() : char
+ setDvdDiscType(dvdDiscType : char) : void
+ getDvdDirector() : string
+ setDvdDirector(dvdDirector : string) : void
+ getDvdRuntime() : double
+ setDvdRuntime(dvdRuntime : double) : void
+ getDvdStudio() : string
+ setDvdStudio(dvdStudio : string) : void
+ getDvdLanguage() : string
+ setDvdLanguage(dvdLanguage : string) : void
+ getDvdSubtitles() : string
+ setDvdSubtitles(dvdSubtitles : string) : void
+ getDvdReleaseDate() : date
+ setDvdReleaseDate(dvdReleaseDate : date) : void
+ getDvdID() : int
+ setDvdID(dvdID : int) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	dvdID	int	N/A	Unique identifier for the DVD
2	dvdDiscType	char	N/A	Represents the type of DVD disc (e.g., Blu-ray, HD-DVD)
3	dvdDirector	string	N/A	The director of the movie or content on the DVD
4	dvdRuntime	double	N/A	The total runtime of the DVD
5	dvdStudio	string	N/A	The studio that produced or released the DVD
6	dvdLanguage	string	N/A	The language of the audio on the DVD
7	dvdSubtitle	string	N/A	The subtitle of the DVD
8	dvdReleaseDate	date	N/A	The release date of the DVD

Table 2. Example of operation design

#	Name	Return type	Description (purpose)

1	getter()	void	Used to retrieve the values of the attributes in the DVD class
2	setter()	v.v	Used to set the values for the attributes in the DVD class

3. Class “Book”

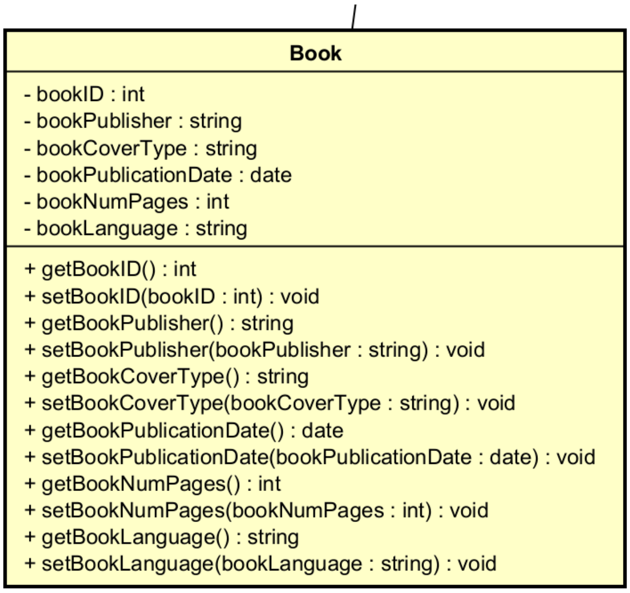

Book
- bookID : int
- bookPublisher : string
- bookCoverType : string
- bookPublicationDate : date
- bookNumPages : int
- bookLanguage : string
+ getBookID() : int
+ setBookID(bookID : int) : void
+ getBookPublisher() : string
+ setBookPublisher(bookPublisher : string) : void
+ getBookCoverType() : string
+ setBookCoverType(bookCoverType : string) : void
+ getBookPublicationDate() : date
+ setBookPublicationDate(bookPublicationDate : date) : void
+ getBookNumPages() : int
+ setBookNumPages(bookNumPages : int) : void
+ getBookLanguage() : string
+ setBookLanguage(bookLanguage : string) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	bookID	int	N/A	Unique identifier for the book
2	bookPublisher	string	N/A	The publisher of the book
3	bookCoverType	string	N/A	The type of cover for the book
4	bookPublicationDate	date	N/A	The date when the book was published
5	bookNumPages	int	N/A	The total number of pages in the book
6	bookLanguage	string	N/A	The language in which the book is written

Table 2. Example of operation design

#	Name	Return type	Description (purpose)

1	getter()	void	Used to retrieve the values of the attributes in the book class
2	setter()	v.v	Used to set the values for the attributes in the book class

4. Class “LP”

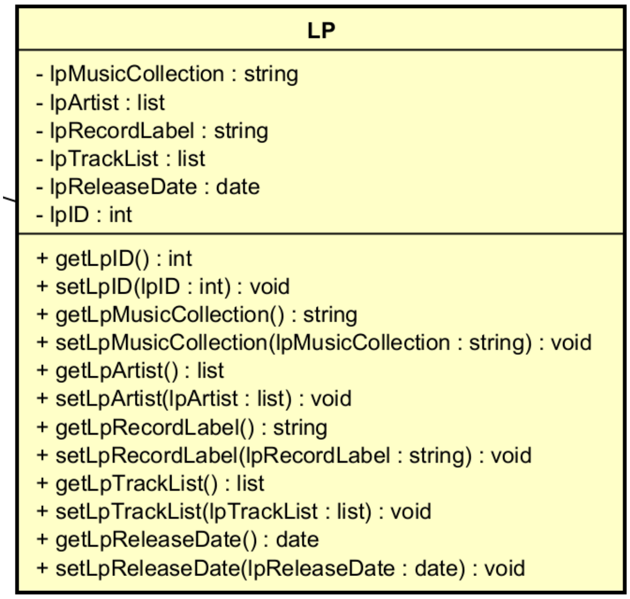

LP
- lpMusicCollection : string
- lpArtist : list
- lpRecordLabel : string
- lpTrackList : list
- lpReleaseDate : date
- lpID : int
+ getLpID() : int
+ setLpID(lpID : int) : void
+ getLpMusicCollection() : string
+ setLpMusicCollection(lpMusicCollection : string) : void
+ getLpArtist() : list
+ setLpArtist(lpArtist : list) : void
+ getLpRecordLabel() : string
+ setLpRecordLabel(lpRecordLabel : string) : void
+ getLpTrackList() : list
+ setLpTrackList(lpTrackList : list) : void
+ getLpReleaseDate() : date
+ setLpReleaseDate(lpReleaseDate : date) : void

Table 1. Example of attribute design

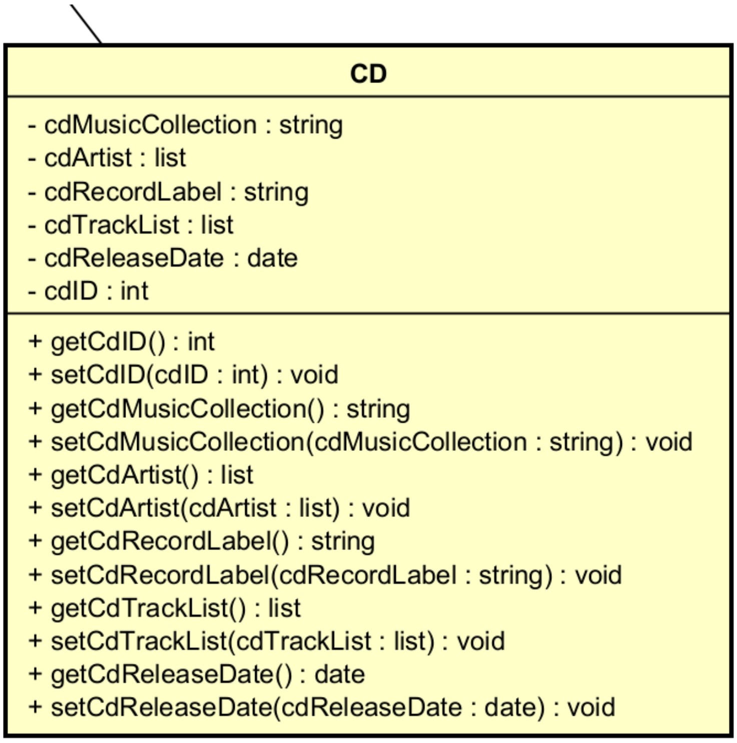
#	Name	Data type	Default value	Description
1	lpID	int	N/A	Unique identifier for the LP
2	lpMusicCollection	string	N/A	Represents the name or type of the music collection in the LP
3	lpArtist	list	N/A	A list of artists associated with the LP
4	lpRecordLabel	string	N/A	The record label that produced or released the LP
5	lpTrackList	list	N/A	A list of tracks (songs) included on the LP
6	lpReleaseDate	date	N/A	he release date of the LP

Table 2. Example of operation design

#	Name	Return type	Description (purpose)

1	getter()	void	Used to retrieve the values of the attributes in the LP class
2	setter()	v.v	Used to set the values for the attributes in the LP class

5. Class “CD”



CD

- cdMusicCollection : string
- cdArtist : list
- cdRecordLabel : string
- cdTrackList : list
- cdReleaseDate : date
- cdID : int

```
+ getCdID() : int
+ setCdID(cdID : int) : void
+ getCdMusicCollection() : string
+ setCdMusicCollection(cdMusicCollection : string) : void
+ getCdArtist() : list
+ setCdArtist(cdArtist : list) : void
+ getCdRecordLabel() : string
+ setCdRecordLabel(cdRecordLabel : string) : void
+ getCdTrackList() : list
+ setCdTrackList(cdTrackList : list) : void
+ getCdReleaseDate() : date
+ setCdReleaseDate(cdReleaseDate : date) : void
```

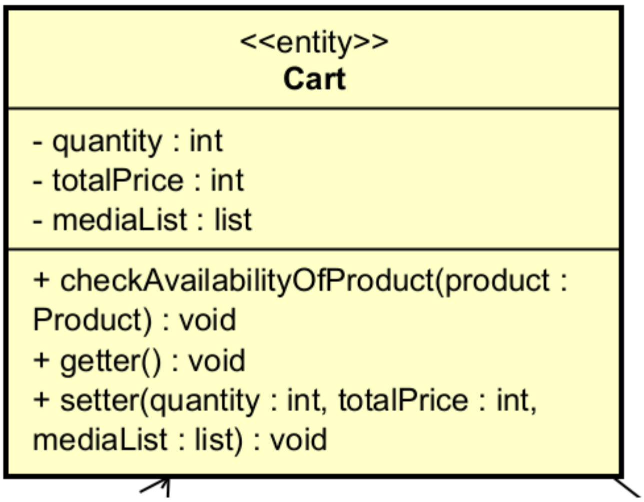
Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	cdID	int	N/A	Unique identifier for the CD
2	cdMusicCollection	string	N/A	Represents the name or type of the music collection in the CD
3	cdArtist	list	N/A	A list of artists associated with the CD
4	cdRecordLabel	string	N/A	The record label that produced or released the CD
5	cdTrackList	list	N/A	A list of tracks (songs) included on the CD
6	cdReleaseDate	date	N/A	he release date of the CD

Table 2. Example of operation design

#	Name	Return type	Description (purpose)
1	getter()	void	Used to retrieve the values of the attributes in the LP class
2	setter()	v.v	Used to set the values for the attributes in the LP class

6. Class “Cart”



<<entity>>
Cart
- quantity : int
- totalPrice : int
- mediaList : list
+ checkAvailabilityOfProduct(product : Product) : void
+ getter() : void
+ setter(quantity : int, totalPrice : int, mediaList : list) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	quantity	int	N/A	The number of items in the shopping cart
2	totalPrice	int	N/A	The total price of the items in the cart
3	mediaList	list	N/A	A list of the products present in the cart

Table 2. Example of operation design

#	Name	Return type	Description (purpose)
1	getter()	void	Used to retrieve the values of the attributes in the Cart class

2	setter()	v.v	Used to set the values for the attributes in the Cart class
3	checkAvailabilityOfProduct()	void	Checks if the product in the cart is available

2. **Method "CheckAvailabilityOfProduct"**

- **Parameter**

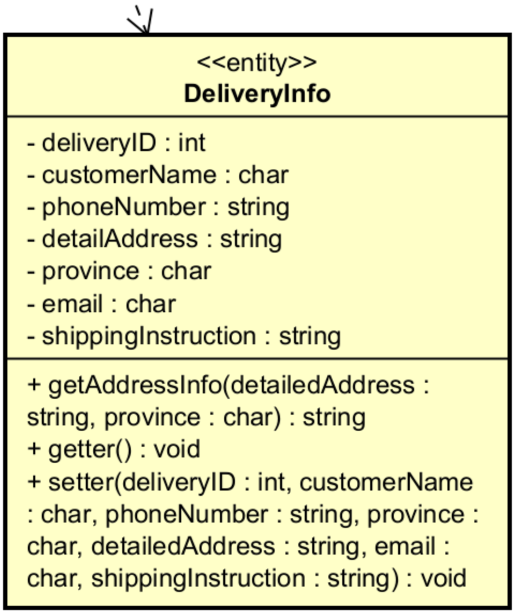
#	Name	Description
1	product: Product	This is an object of the Product class that represents a specific product to check for availability

- **Exception**

#	Name	Description
1	Null Pointer Exception	If the product passed to the method is null, trying to access any of its properties would throw a NullPointerException
2	Invalid Quantity	If the quantity in the cart is negative or invalid, we could throw an exception to handle such cases.
3	Product Availability Logic	If for some reason the mediaStock of the product is not a valid number (negative, for instance), we could throw an exception

- **How to use parameter/attribute:** *checkAvailabilityOfProduct()* uses the product parameter to check if the inventory stock is sufficient for the customer's order.

7. Class "DeliveryInfo"



<<entity>>
DeliveryInfo

- deliveryID : int
- customerName : char
- phoneNumber : string
- detailAddress : string
- province : char
- email : char
- shippingInstruction : string

+ getAddressInfo(detailedAddress : string, province : char) : string
+ getter() : void
+ setter(deliveryID : int, customerName : char, phoneNumber : string, province : char, detailedAddress : string, email : char, shippingInstruction : string) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	deliveryID	int	N/A	Unique identifier for the delivery
2	customerName	char	N/A	Name of the customer
3	phoneNumber	string	N/A	Phone number of the customer
4	detailAddress	string	N/A	Detailed address for the delivery (e.g., street address)
5	province	char	N/A	Address for the delivery
6	email	string	N/A	Email of the customer
7	shippingInstruction	string	N/A	Any special instructions for the delivery

Table 2. Example of operation design

#	Name	Return type	Description (purpose)
1	getter()	void	Used to retrieve the values of the attributes in the DeliveryInfo class
2	setter()	v.v	Used to set the values for the attributes in the DeliveryInfo class
3	getAddressInfo()	string	Method to get the full address by combining the detailed address and province

1. **Method “getAddressInfo”**

- **Parameter**

#	Name	Description
1	detailedAddress: string	Detailed address for the delivery (e.g., street address)
2	province: char	Address for the delivery

- **Exception: N/A**
- **How to use parameter/attribute:** *getAddressInfo()* uses the parameters to form a complete address for the customer.

8. Class “RushOrderForm”

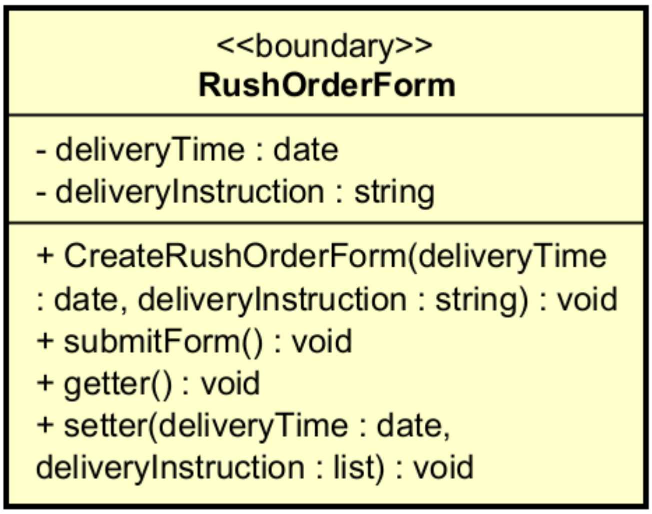

<<boundary>>
RushOrderForm
- deliveryTime : date
- deliveryInstruction : string
+ CreateRushOrderForm(deliveryTime : date, deliveryInstruction : string) : void
+ submitForm() : void
+ getter() : void
+ setter(deliveryTime : date, deliveryInstruction : list) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description
1	deliveryTime	date	N/A	The date and time when the delivery is to occur
2	deliveryInstruction	string	N/A	Instructions for the delivery

Table 2. Example of operation design

#	Name	Return type	Description (purpose)
1	getter()	void	Used to retrieve the values of the attributes in the RushOrderForm class
2	setter()	v.v	Used to set the values for the attributes in the RushOrderForm class
3	CreateRushOrderForm()	void	A method to create or initialize the rush order form
4	submitForm()	void	Submits the form with the delivery time and instructions provided by the customer

1. Method “submitForm”

- **Parameter**

#	Name	Description

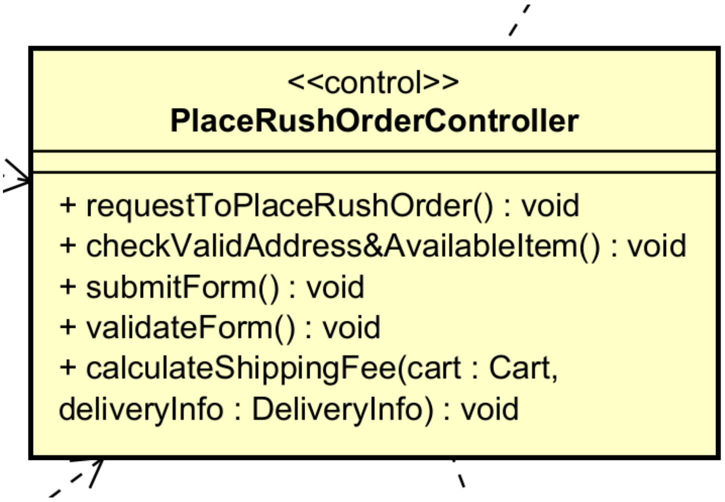
1	deliveryTime: date	
2	deliveryInstruction: string	

- **Exception**

#	Name	Description
1	IllegalArgumentException	If parameter is not valid, it might not be acceptable
2	InvalidDateException	If the deliveryTime is in the past, it should be considered invalid
3	NullPointerException	If deliveryTime or deliveryInstruction is null, it could lead to errors when trying to use them

- **How to use parameter/attribute:** *submitForm()* receives the parameters entered by the user to set up the rush order information

9. Class "PlaceRushOrderController"



<<control>>
PlaceRushOrderController
+ requestToPlaceRushOrder() : void
+ checkValidAddress&AvailableItem() : void
+ submitForm() : void
+ validateForm() : void
+ calculateShippingFee(cart : Cart, deliveryInfo : DeliveryInfo) : void

Table 1. Example of attribute design

#	Name	Data type	Default value	Description

Table 2. Example of operation design

#	Name	Return type	Description (purpose)
1	requestToPlaceRushOrder()	void	Requests to place a rush order

2	checkValidAddress&AvailableItem()	void	Checks if the delivery address supports rush order service and if the items in the order are eligible for rush order delivery
3	submitForm()	void	Submits the order form with delivery and payment details for processing
4	validateForm()	void	Validates the form fields to ensure that all required fields are filled and the information is correct
5	calculateShippingFee()	void	Calculates the shipping fee based on the delivery location, order weight, and whether the rush order delivery option is selected

1. **Method "validateForm"**

- **Parameter**

#	Name	Description
1	rushOrderForm: RushOrderForm	This is an object of the RushOrderForm class that represents a specific RushOrderInfo for checking validate information

- **Exception**

#	Name	Description
1	InvalidForm	If the information from RushOrderForm is invalid, it could lead to errors when trying to use them

- **How to use parameter/attribute:** *validateForm()* uses the RushOrderForm parameter to check if the additional information is valid or not.

2. **Method "calculateShippingFee"**

- **Parameter**

#	Name	Description
1	cart: Cart	This is an object of the Cart class that represents a specific cart for calculating shipping fee
2	deliveryInfo: DeliveryInfo	This is an object of the DeliveryInfo class that represents a specific DeliveryInfo for calculating shipping fee

- **Exception: N/A**
- **How to use parameter/attribute:** *calculateShippingFee()* uses parameters to calculate shipping fee.

10. Class “PlaceOrderController”

Table 1. Example of attribute design

#	Name	Data type	Default value	Description

Table 2. Example of operation design

#	Name	Return type	Description (purpose)