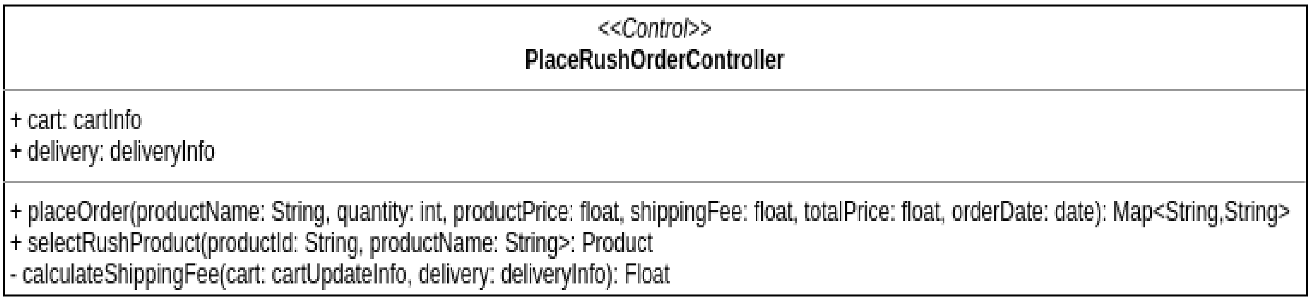
Class Design For Place Rush Order



<<Control>>
PlaceRushOrderController

+ cart: cartInfo
+ delivery: deliveryInfo

+ placeOrder(productName: String, quantity: int, productPrice: float, shippingFee: float, totalPrice: float, orderDate: date): Map<String,String>
+ selectRushProduct(productId: String, productName: String>: Product
- calculateShippingFee(cart: cartUpdateInfo, delivery: deliveryInfo): Float

Attribute

#	Name	Data type	Default value	Description
1	cart	cartInfo	NULL	Represent the cart used for place rush order
2	delivery	deliveryInfo	NULL	Represent the delivery user used for place rush order

Operation

#	Name	Return type	Description
1	placeOrder	Map<String,String>	Represent the cart used for place rush order
2	selectRushProduct	Product	Choose rush product (If the product can rush order)

Parameter:

- productName: The name of product
- productId: The id of product
- quantity: The quantity of product
- productPrice: The price of product
- shippingFee: The fee of order
- totalPrice: The total price of order
- orderDate: The order date

Exception:

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

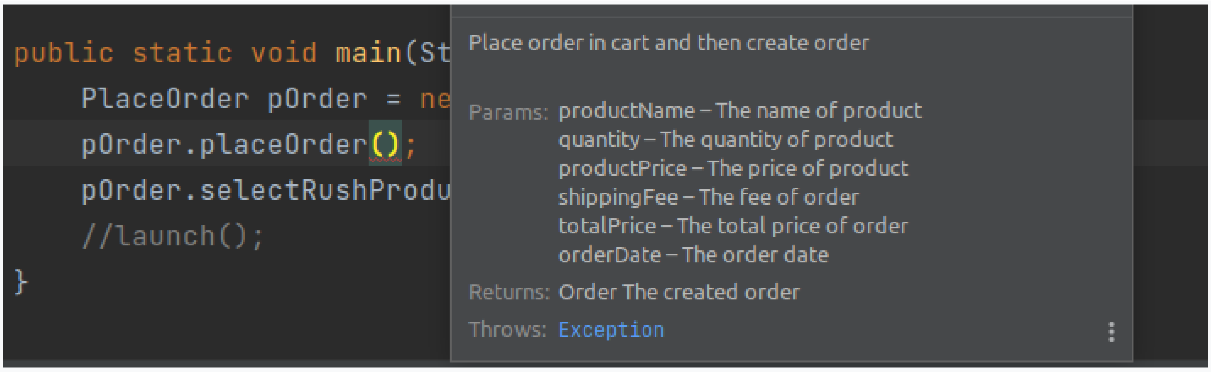
Method:

None

State:

None

Illustration:



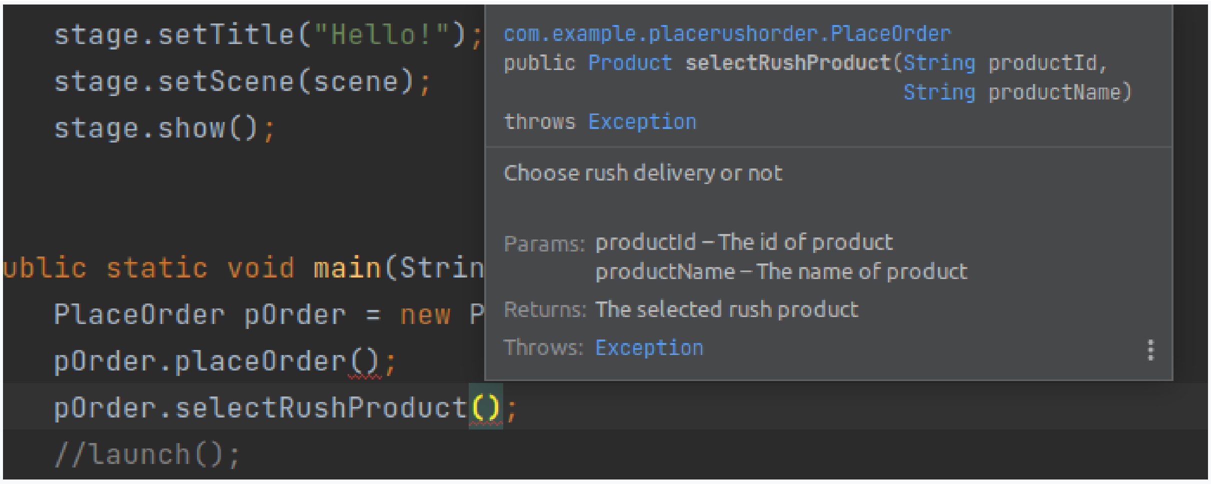
```
public static void main(St
    PlaceOrder pOrder = ne
    pOrder.placeOrder();
    pOrder.selectRushProdu
    //launch();
}
```

Place order in cart and then create order

Params: productName – The name of product
quantity – The quantity of product
productPrice – The price of product
shippingFee – The fee of order
totalPrice – The total price of order
orderDate – The order date

Returns: Order The created order

Throws: Exception



```
    stage.setTitle("Hello!");
    stage.setScene(scene);
    stage.show();


ublic static void main(Strin
    PlaceOrder pOrder = new P
    pOrder.placeOrder();
    pOrder.selectRushProduct();
    //launch();
```

```
com.example.placerushorder.PlaceOrder
public Product selectRushProduct(String productId,
                                 String productName)
throws Exception
```

Choose rush delivery or not

Params: productId – The id of product
productName – The name of product

Returns: The selected rush product

Throws: Exception