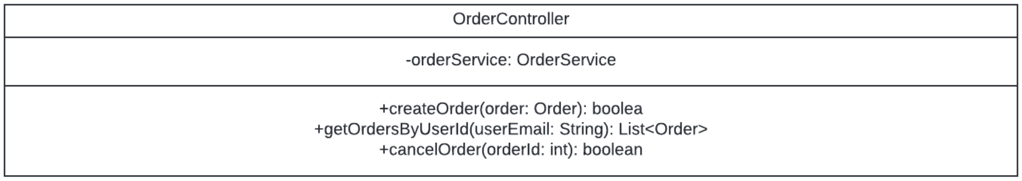

OrderController
-orderService: OrderService
+createOrder(order: Order): boolea
+getOrdersByUserId(userEmail: String): List<Order>
+cancelOrder(orderId: int): boolean

## Invariant: -

- context OrderController

  inv : self.orderService <> null

## Pre-Condition: -

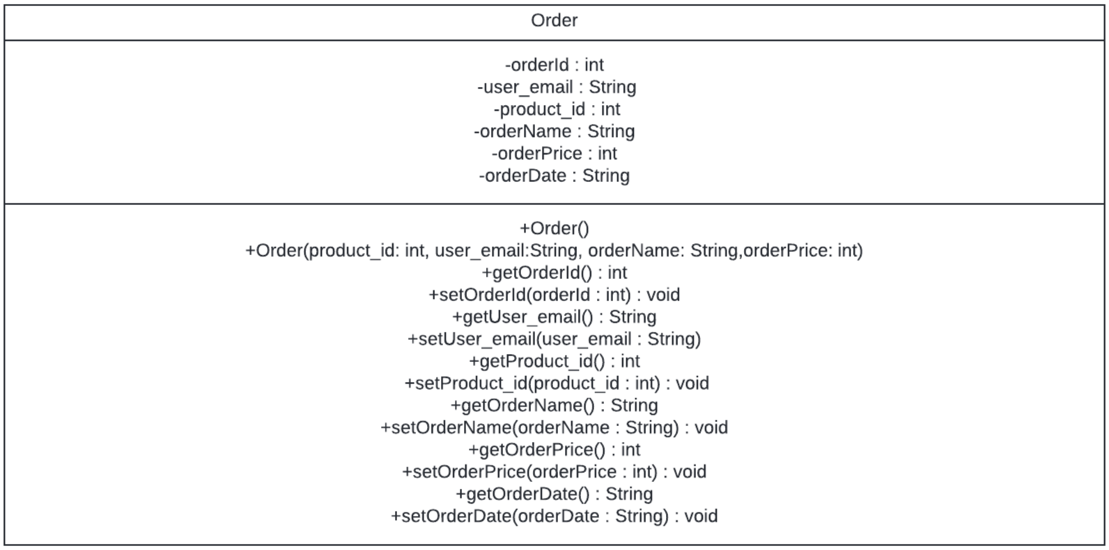
- context OrderController::getOrdersByUserId(userEmail : String)

  pre: userEmail.matches('^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.com$')

- context OrderController::createOrder(order : Order)
  pre: order <> null
- context OrderController::cancelOrder(orderId : int)
  pre: orderId > 0

## Post- Condition: -

- context OrderController::createOrder(order : Order) :
  post: result <> null
- context OrderController::getOrdersByUserId(userEmail : String) :
  post: result <> null
- context OrderController::cancelOrder(orderId : int) :
  post: result <> null



Order

-orderId : int
-user_email : String
-product_id : int
-orderName : String
-orderPrice : int
-orderDate : String

+Order()
+Order(product_id: int, user_email:String, orderName: String,orderPrice: int)
+getOrderId() : int
+setOrderId(orderId : int) : void
+getUser_email() : String
+setUser_email(user_email : String)
+getProduct_id() : int
+setProduct_id(product_id : int) : void
+getOrderName() : String
+setOrderName(orderName : String) : void
+getOrderPrice() : int
+setOrderPrice(orderPrice : int) : void
+getOrderDate() : String
+setOrderDate(orderDate : String) : void

## Invariant: -

- context Order
  inv:Order.allInstances()->isUnique(orderId)
- context Order
  inv: self.user_email.includes('@') and self.user_email.size() > 5
- context Order
  inv:self.orderPrice > 0
- context Order
  inv:self.orderDate.matches('^\d{4}-\d{2}-\d{2}$')

## Pre-Condition: -

- context Order::setOrderId(orderId: Int)
  pre: orderId > 0
- context Order::setUser_email(user_email: String)
  pre: user_email.includes('@') and user_email.size() > 5
- context Order::setOrderName(orderName: String)
  pre: not orderName.oclIsUndefined() and orderName.size() > 0

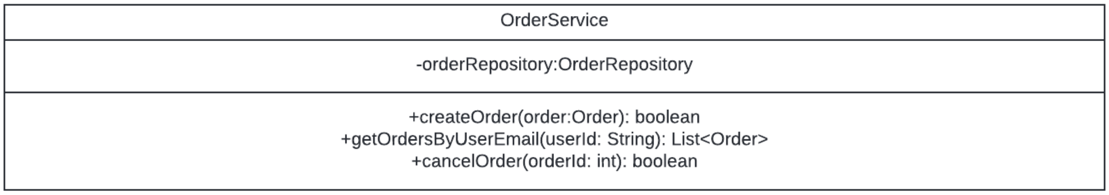
## Post- Condition: -

- context Order::getOrderId(): int
  post: result = self.orderId
- context Order::getUser_email(): String
  post: result = self.user_email

- context Order::getProduct_id(): int

post: result = self.product_id

- context Order::setOrderName(orderName : String)
  post: self.orderName = orderName
- context Order::getOrderName(): String
  post: result = self.orderName
- context Order::getOrderDate(): String
  post: result = self.orderDate



OrderService

-orderRepository:OrderRepository

+createOrder(order:Order): boolean
+getOrdersByUserEmail(userId: String): List<Order>
+cancelOrder(orderId: int): boolean

## **<u>Invariant: -</u>**

- context OrderService
  inv validOrderRepository:self.orderRepository <> null

## **<u>Pre-Condition: -</u>**

- context OrderService::createOrder(order : Order) : boolean
  pre: order <> null
- context OrderService::getOrdersByUserEmail(userId : String) : List<Order>
  pre: userId <> null and userId.size() > 0
- context OrderService::cancelOrder(orderId : int) : boolean
  pre: orderId > 0

## **<u>Post- Condition: -</u>**

- context OrderService::createOrder(order : Order) : boolean
  post: result = true implies (order.orderDate <> null and order.orderDate = DateTime.now().toString('yyyy-MM-dd'))
- context OrderService::getOrdersByUserEmail(userId : String) : List<Order>
  post: result <> null implies result->forAll(o | o.user_email = userId)
- context OrderService::cancelOrder(orderId : int) : boolean
  post: result = true implies not OrderRepository.allInstances()->exists(o | o.orderId = orderId)