1. PAYORDERMANAGER

* Class Name: PayOrderManager
* Class Description: The PayOrderManager class orchestrates the process of initiating payment orders within the system. It oversees the handling of payment requests, ensuring that transactions are executed smoothly and securely. This class plays a pivotal role in managing the flow of payment-related operations, facilitating seamless interactions between the application and the payment gateway.
* Operation Definition:
  + payOrder(amount: double, orderInfo: String, client: Client): void
    - Description: initiates a payment transaction with the specified details. It interacts with the underlying payment gateway to process the transaction securely. Upon successful completion, this method does not return any value (void).
    - Parameters:
      * @param amount A double number representation of the amount to be paid in the transaction.
      * @param orderInfo A string representation of essential detailed about the order being processed.
      * @param client The client initiating the payment order transaction, providing necessary identification and contact information
      * @return void
  + payOrder(queryString: String): void
    - Description: Processes a payment order based on the provided query string. This method may involve parsing and interpreting the query parameters to determine the appropriate action to take. It handles various types of payment requests and ensures that they are executed in accordance with the system's requirements.
    - Parameters:
      * @param queryString The query string containing result and info of transaction.
      * @return void

1. REQUEST

* Class Name: Request
* Class Description: The Request class represents a request for initiating a payment transaction within the system. It encapsulates the necessary data and functionality required to create and process payment requests. This class serves as an intermediary between the application layer and the payment gateway, facilitating the generation of payment orders and the construction of query URLs for communication with external systems.
* Operation Definition:
  + Request(amount: double, orderInfo: String): void
    - Description: creates a payment request object with the provided transaction details, including the payment amount and order information. It prepares the request for further processing, such as validation and submission to the payment gateway
    - Parameters:
      * @param amount A double number representation of the amount to be paid in the transaction.
      * @param orderInfo A string representation of essential detailed about the order being processed.
      * @return void
  + buildQueryURL(): void
    - Description: generates a query URL based on the payment request parameters. It assembles the necessary query parameters, such as the payment amount and order information, into a URL format suitable for communication with external systems, such as the payment gateway. This method prepares the request for transmission and ensures that it contains all the essential information required for processing
    - Parameters:
      * @return void

1. RESPONSE

* Class Name: Response
* Class Description: The Response class represents a response received from the payment gateway after processing a payment request. It encapsulates the functionality required to parse the response data, handle error codes, and extract relevant information from the response. This class serves as a crucial component in managing the response received from external systems, ensuring that the payment transaction is processed accurately and efficiently.
* Operation Definition:
  + Response(queryString: String): void
    - Description: constructs a response object based on the query string received from the payment gateway. It initializes the response object with the provided data, preparing it for further processing and analysis. This method does not return any value (void).
    - Parameters:
      * @param queryString The string containing information about the response received from the payment gateway.
      * @return void
  + parseQueryString (): void
    - Description: parses the query string received in the response to extract essential information, such as transaction status, payment confirmation codes, and error codes. It processes the response data to ensure that the relevant information is captured accurately for further analysis and handling. This method does not return any value (void).
    - Parameters:
      * @return void
  + handleErrorCode(): void
    - Description: checks for any error codes present in the response and takes appropriate action based on the error code received. It facilitates error handling by identifying and addressing any issues that may have occurred during the payment transaction. This method ensures that errors are handled effectively, minimizing disruption to the payment processing workflow. This method does not return any value (void).
    - Parameters:
      * @return void

1. CLIENT

* Class Name: Client
* Class Description: The Client class represents a client or user interacting with the payment system. It encapsulates the functionality for updating transaction status based on the outcome of payment processing, whether successful or unsuccessful. This class serves as a key component in managing client-related actions and interactions within the payment system.
* Operation Definition:
  + updateTransactionOnFailure(exception: Exception): void
    - Description: updates the transaction status in the event of payment failure. It receives an exception object representing the error or failure encountered during payment processing and takes appropriate action to update the transaction status accordingly. This method ensures that failed transactions are handled appropriately, and relevant information about the failure is recorded for further analysis or follow-up actions.
    - Parameters:
      * @param exception The exception object containing details about the error or failure encountered during payment processing.
      * @return void
  + updateTransactionOnSuccess(trans: Transaction): void
    - Description: updates the transaction status upon successful payment processing. It receives a transaction object representing the successfully processed payment transaction and updates the transaction status to reflect the successful completion of the transaction. This method ensures that successful transactions are accurately recorded and appropriately reflected in the system, providing users with confirmation of successful payment processing.
    - Parameters:
      * @param trans The transaction object containing details about the successfully processed payment transaction
      * @return void

1. VNPAYDISPLAY

* Class Name: VNPayDisplay
* Class Description: The VNPayDisplay class manages the display and interaction functionalities related to VNPay payment orders within the system. It encapsulates methods for sending payment orders, displaying URLs, checking for URL changes, and closing the display interface. This class facilitates the user interface aspects of VNPay payment processing, ensuring smooth communication and interaction between the application and the VNPay payment gateway.
* Operation Definition:
  + sendPayOrder(queryURL: String): void
    - Description: initiates the sending of a payment order to VNPay using the provided query URL. It communicates with the VNPay payment gateway to process the payment order securely and efficiently. This method ensures that the payment order is transmitted to VNPay for further processing, facilitating the completion of the payment transaction.
    - Parameters:
      * @param queryURL (String): A string containing information about the payment order.
      * @return void
  + displayURL(url: String): void
    - Description: presents the provided URL to the user for interaction. It displays the URL in a user-friendly manner, allowing the user to view and interact with the content as needed. This method ensures that relevant URLs, such as payment confirmation pages or redirect URLs, are presented to the user for further action or verification.
    - Parameters:
      * @param url The String URL to be displayed to the user.
      * @return void
  + checkURLChange(): void
    - Description: monitors changes in the displayed URL to detect any redirection or navigation events. It ensures that the user is kept informed by check change in the URL displayed on the interface, allowing them to track the progress of the payment transaction or navigate to relevant pages as necessary.
    - Parameters:
      * @return void
  + close(): void
    - Description: terminates the VNPay display interface, closing any active windows or interfaces associated with VNPay display functionalities. It ensures that the display interface is properly closed and any resources used for display are released, providing a clean and seamless user experience.
    - Parameters:
      * @return void

1. TRANSACTIONINFO

* Class Name: TransactionInfo
* Class Description: The TransactionInfo class represents information about a transaction, encapsulating data received in a response from the payment gateway. It provides functionality for initializing transaction information based on the response received from the payment gateway.
* Operation Definition:
  + ransactionInfo(response: Response): void
    - Description: initializes transaction information based on the data received in the response object from the payment gateway. It extracts relevant transaction details, such as transaction ID, payment status, and other metadata, from the response and populates the TransactionInfo object with this information. This method ensures that transaction information is accurately recorded and available for further processing or display within the system.
    - Parameters:
      * @param response The object containing data received from the payment gateway.
      * @return void