How to use:

Method 1:

1. Import library arimaModel in your script.

```
require_once __DIR__ . '/../src/arimaModel.php';
```

2. Prepare your time series to an ordered “based on time” array .

```
$data = array(
    266.0,
    145.9,
    183.1,
    119.3,
    180.3,
    168.5,
    231.8,
    224.5,
    192.8,
    122.9,
    336.5,
    185.9,
    194.3,
    149.5,
    210.1,
    273.3,
    191.4,
    287.0,
    226.0,
    303.6,
    289.9
);
```

3. To use arima model set the order array “array(p,d,q);”.

```
$order = array(1,1,1);
```

And call the arima function with the parameters (data , order , pred_num=1)
Pred_num is the number of predictions you want , the default value is 1.

```
$res = arimaModel::arima($data , $order);
```

The result “$res”, is an ordered array of predictions.

4. To use auto arima model call the auto_arima function with the parameters
   (data , pred_num=1 , algo = “AIC”)
   Pred_num is the number of predictions you want, the default value is 1.
   algo parameter is the algorithm on witch the parameters has been chosen , it can take this values (“AIC”,”BIC”,”AIC+BIC”,”MSE”) , default is “AIC”.

   ```
   $res= arimaModel::auto_arima($data);
   ```

   The result $res is an order array of predictions.

5. To get the order of the auto arima model you can call get_auto_arima_order() with no parameters , but you must’ve called the auto_arima before calling this function.

   ```
   $pred_order = arimaModel::get_auto_arima_order();
   ```

   The result $pred_order is an array of parameters “p,d,q”, you can access them by name.

   ```
   echo $pred_order[“p”];
   ```

Method 2:

1. Import library arima controllers in your script.

   ```
   require_once __DIR__ . '/../src/arimaContrller.php';
   ```

   Or

   ```
   require_once __DIR__ . '/../src/autoArimaContrller.php';
   ```

2. Create controllers objects

   ```
   $order = array(0,1,1); // array(p,d,q)
   $arc = new arimaContrller($order);
   ```

   Or

   ```
   $arc = new autoArimaContrller("BIC"); // use BIC algorithm
   ```

3. Get data using query , the query most return two columns (date,value) , and you must insert the names of the columns to the function calling them so that arima can handle timeseries based on date column and predict values based on value column , there is two way to enter a query :

a. Insert query as string:

```
$res = $arc->setDataBaseConnection($link)->query($sql,"date" ,"value" )->forecast(1);
```

setDataBaseConnection

$link : is the connection to data base

Query

$sql : is the sql query
“date” : is the name of the date column
“value” : is the name of the value column

Or

```
$res = $arc->setDataBaseConnection($link)->query($sql,"date" ,"value" ,"01h")->forecast(1);
```

Query

“01h” : is the step parameter default is null , if entered then the data returned by the query most meet the conditions:

- Data is sorted by dates.
- Step between dates most equal the step parameter if there is missing dates it will replace it with the date missing and the value null.

This **parameter most take two digit and a char** ,the digit is for the duration and the char is one of this values (Y: year , m: month , d: day , h: hour , i: minutes , s: seconds)

b. Use built in functions to build query:

```
$res = $arc->setDataBaseConnection($link)->select(date , 'value','table_name’)->where("condition ")->groupBy("column")->orderBy("date",true)->forecast(1);
```

setDataBaseConnection

$link : is the connection to data base

Select

“date” : is the name of the date column

"value" : is the name of the value column
"table_name" : is the name of the table

Where

"condition" : is the a condition to filter the data

groupBy

"column" : column name to group data by

orderBy

"date" : is the name of the column you want to sort data by its values
true : meaning descending=true witch sort data DESCENDING
if false it will order data ASCENDING

"you can test all functions from tests folder"