**package** agile\_project;

**import** org.junit.\*;

**import** **static** org.junit.Assert.*assertEquals*;

**import** **static** org.junit.Assert.*assertNotNull*; // Import for assertNotNull

**import** **static** org.junit.Assert.*fail*;

**public** **class** OrderTests {

// TestNumber:1

// Objective: Verify type(daily) is assigned to order

// Input: daily

// Output: True

@Test

**public** **void** testcase001() **throws** codeException{

Order type = **new** Order("daily","date","type");

**try** {

**boolean** result = type.validType("daily");

*assertEquals*(**true**,result);

}**catch**(codeException e){

*fail*("exception not expected");

}

}

// TestNumber:2

// Objective: Verify type weekly is assigned to order

// Input: weekly

// Output: True

@Test

**public** **void** testcase002() **throws** codeException{

Order type = **new** Order("daily","date","type");

**try** {

**boolean** result = type.validType("weekly");

*assertEquals*(**true**,result);

}**catch**(codeException e){

*fail*("exception not expected");

}

}

// TestNumber:3

// Objective: Verify type monthly is assigned to order

// Input: monthly

// Output: True

@Test

**public** **void** testcase003() **throws** codeException{

Order type = **new** Order("type","date","title");

**try** {

**boolean** result = type.validType("monthly");

*assertEquals*(**true**,result);

}**catch**(codeException e){

*fail*("exception not expected");

}

}

// TestNumber:4

// Objective: Verify type monthly is assigned to order

// Input: monthly

// Output: True

@Test

**public** **void** testcase004() **throws** codeException{

Order date = **new** Order("title","type","date");

**try** {

**boolean** result = date.validDate("Current");

*assertEquals*(**true**,result);

}**catch**(codeException e){

*fail*("exception not expected");

}

}

// TestNumber:5

// Objective: Verify type monthly is assigned to order

// Input: monthly

// Output: True

@Test

**public** **void** testcase005() **throws** codeException{

Order title = **new** Order("title","type","date");

**try** {

**boolean** result = title.validDate("");

*assertEquals*(**true**,result);

}**catch**(codeException e){

*fail*("exception not expected");

}

}

// @Test

// public void testStartDateGeneration() {

// Order order = new Order();

// order.generateStartDate();

//

// assertNotNull(order.getStartDate());

// }

//

// @Test

// public void testOrderTitleCreation() {

// Order order = new Order();

// order.createOrderTitle();

//

// assertNotNull(order.getOrderTitle());

// }

}

**package** agile\_project;

**public** **class** Order {

**private** String type;

**private** String date;

**private** String title;

**public** Order (String type, String date ,String title) **throws** codeException {

**throw** **new** codeException("Constructor not yet implemented");

}

**public** **void** givePermission() **throws** codeException {

**throw** **new** codeException("Method not yet implemented");

}

**public** **boolean** validType (String type) **throws** codeException{

**boolean** result = **true**;

**if**(type.equals("daily")) {

result = **true**;

}

**else** **if**(type.equals("weekly")) {

result = **true**;

}

**else** **if**(type.equals("monthly")) {

result = **true**;

}

**else** {

result = **false**;

}

**return** result;

}

// public void setOrderType(String orderType) {

// this.orderType = orderType;

// }

//

// public String getOrderType() {

// return orderType;

// }

//

// public void generateStartDate() {

// // Add logic to generate the start date

// // For example, you can set it to the current date as a placeholder

// this.startDate = "2023-10-23";

// }

//

// public String getStartDate() {

// return startDate;

// }

//

// public void createOrderTitle() {

// // Add logic to create the order title

// // For example, you can set it to a fixed value as a placeholder

// this.orderTitle = "Sample Order Title";

// }

//

// public String getOrderTitle() {

// return orderTitle;

// }

**public** **boolean** validDate (String date) **throws** codeException{

**boolean** result = **true**;

**if**(type.equals("Current")) {

result = **true**;}

**else** {

result = **false**;

}

**return** result;

}

}

**package** agile\_project;

**public** **class** codeException **extends** Exception{

String message;

**public** codeException(String e) {

message = e;

}

**public** String getMessage() {

**return** message;

}

}