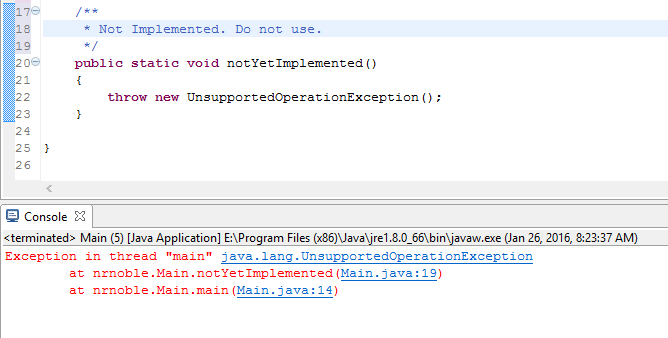
Neal Noble

IT301

Exercises - Throwing Exceptions (Perspective A)

January 27nd, 2016

1. Write a method called public void notYetImplemented(). The method should throw an UnsupportedOperationException when the method is called. Call your method and submit a screenshot of the error message you get on the Java console.



2. Write a program that asks the user on the Java console for two numbers "a" and "b" and prints the result of a / b. Your program should throw an ArithmeticException, with an appropriate message, if the user tries to divide by zero.

**public** **static** **int** division (**int** \_a, **int** \_b)

{

**if** (\_a == 0 ||\_b == 0)

{

ArithmeticException exception = **new** ArithmeticException("Division by zero");

**throw** exception;

}

**return** \_a / \_b;   
}

3. Write a BankAccount class that stores a bank account id (a number between 1000-9999), an account holders first/last name and an account balance. Your class should have a constructor that passes in all four field values. For each of the following bad inputs to your constructor, throw the exception type listed:

|  |  |
| --- | --- |
| **Input** | **Exception type** |
| Account id outside the range 1000-9999 | RuntimeException |
| First name or last name null or empty | NullPointerException |
| A negative account balance | IllegalStateException |

**public** **static** **void** BankAccount (**int** \_accountID, String \_firstName,

String \_lastName, **int** \_accountBalance)

{

**if** (\_accountID < 1000 || \_accountID > 9999)

{

**throw** **new** RuntimeException ("Account number out of range");

}

**if** ((\_firstName == "") || (\_firstName == **null**))

{

**throw** **new** NullPointerException ("First name can not be empty");

}

**if** ((\_lastName == "") || (\_lastName == **null**))

{

**throw** **new** NullPointerException ("Last name can not be empty");

}

**if** (\_accountBalance < 0)

{

**throw** **new** IllegalStateException ("Account balance

can not be negitive");

}

}