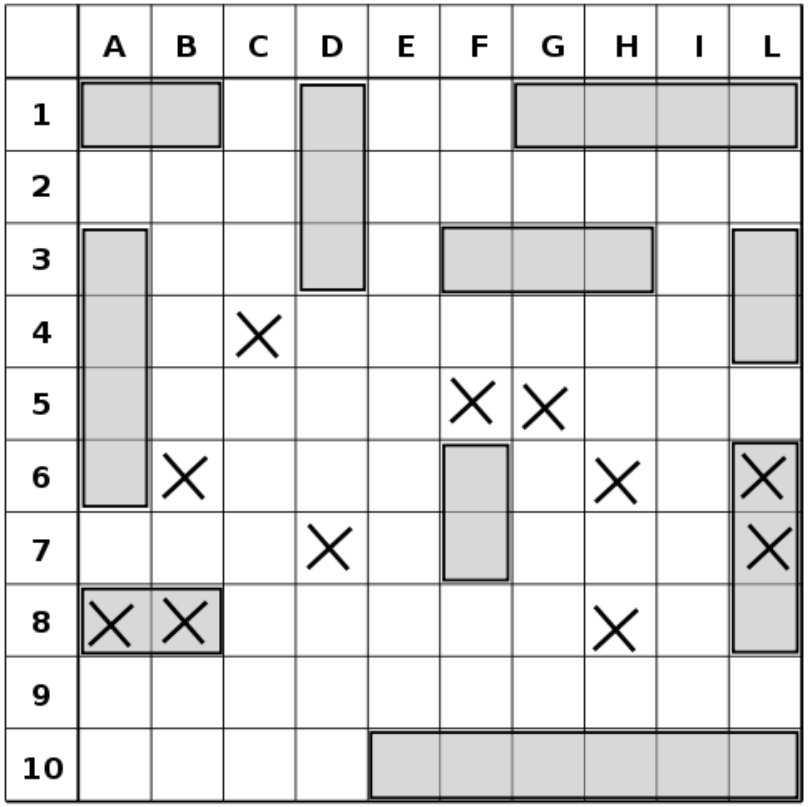
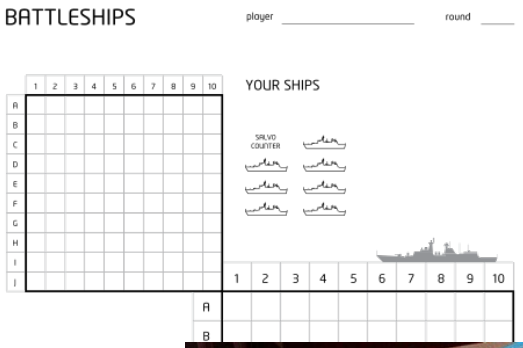
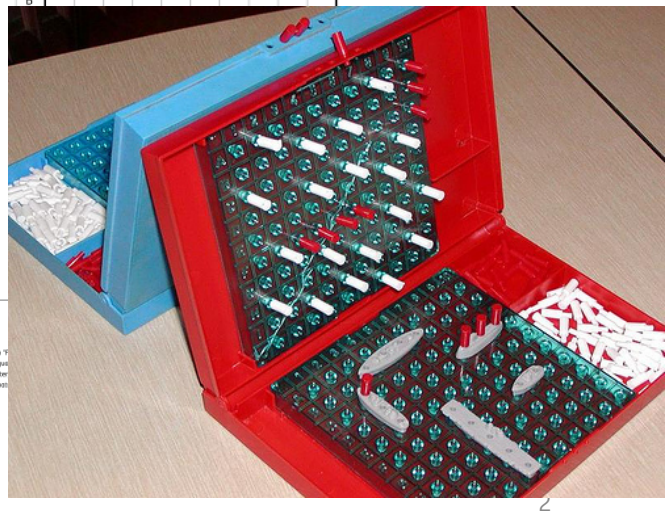
Writing a Program: Sink a .Com

Sink a .Com (Battleship)

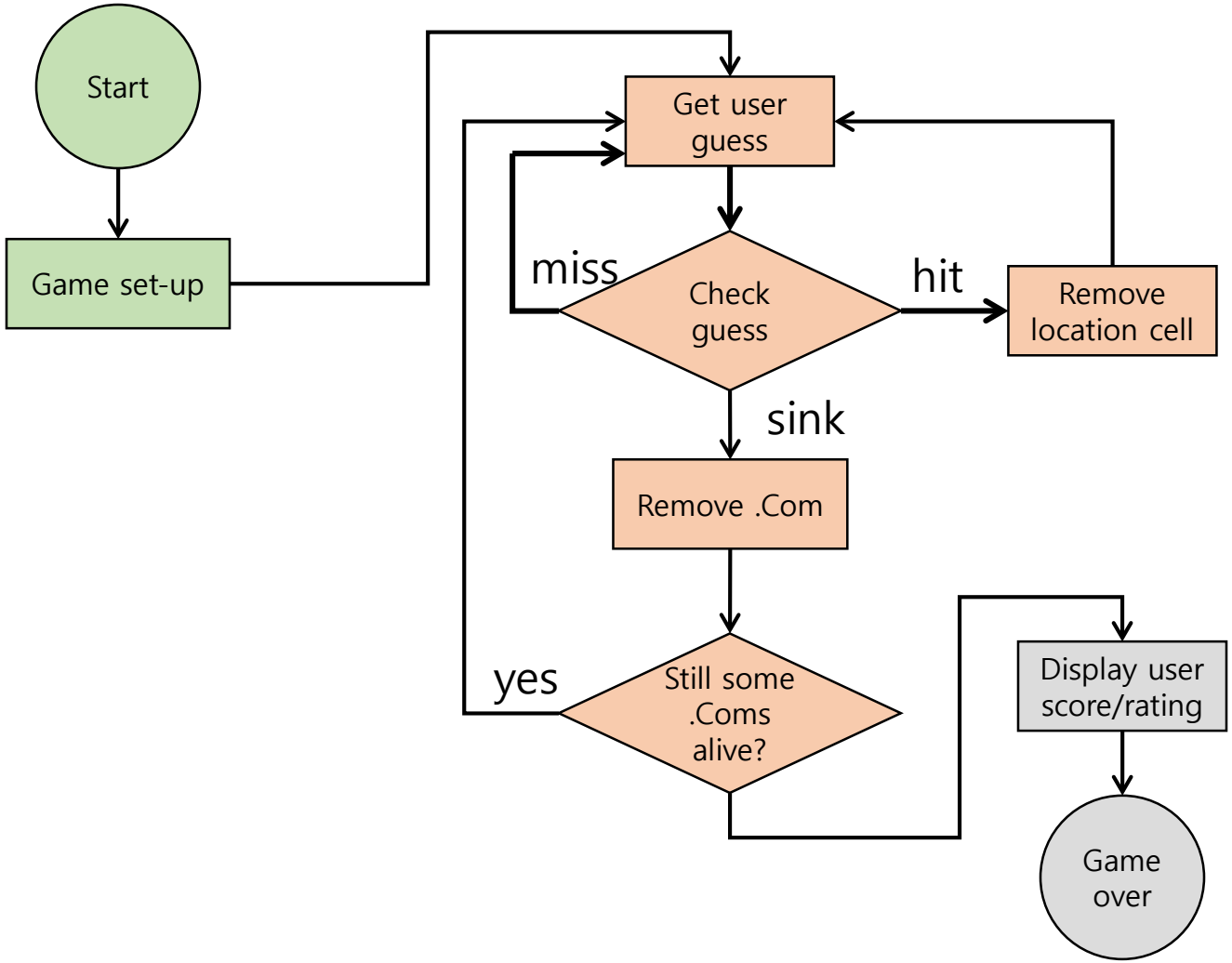
- Battleship (Game): a guessing game for two players







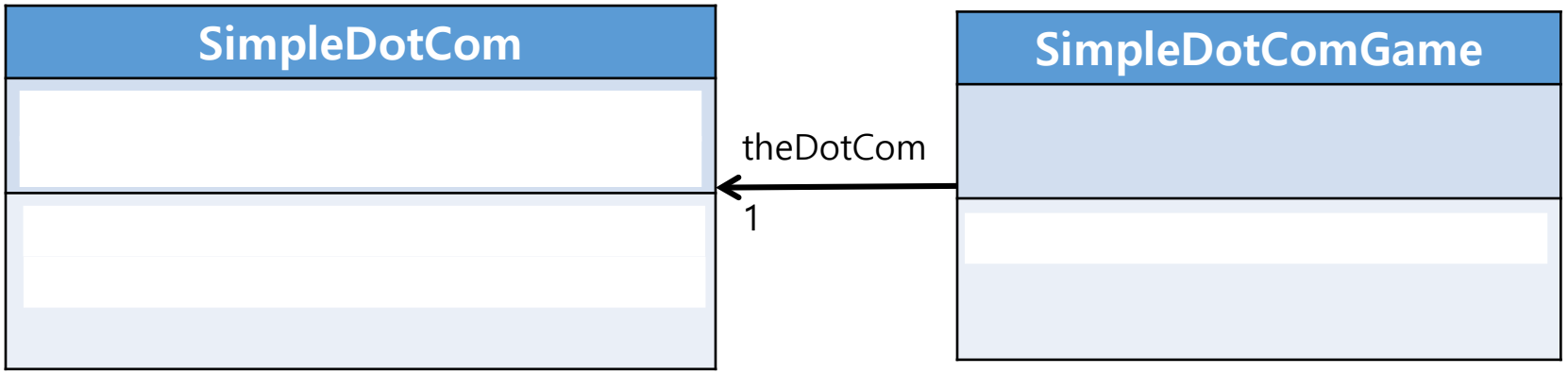
Sink a .Com: Flow



Sink a .Com 1D

- Simple .Com game

- Class diagram
  - The relationship between the classes will be defined later.



Developing a Class

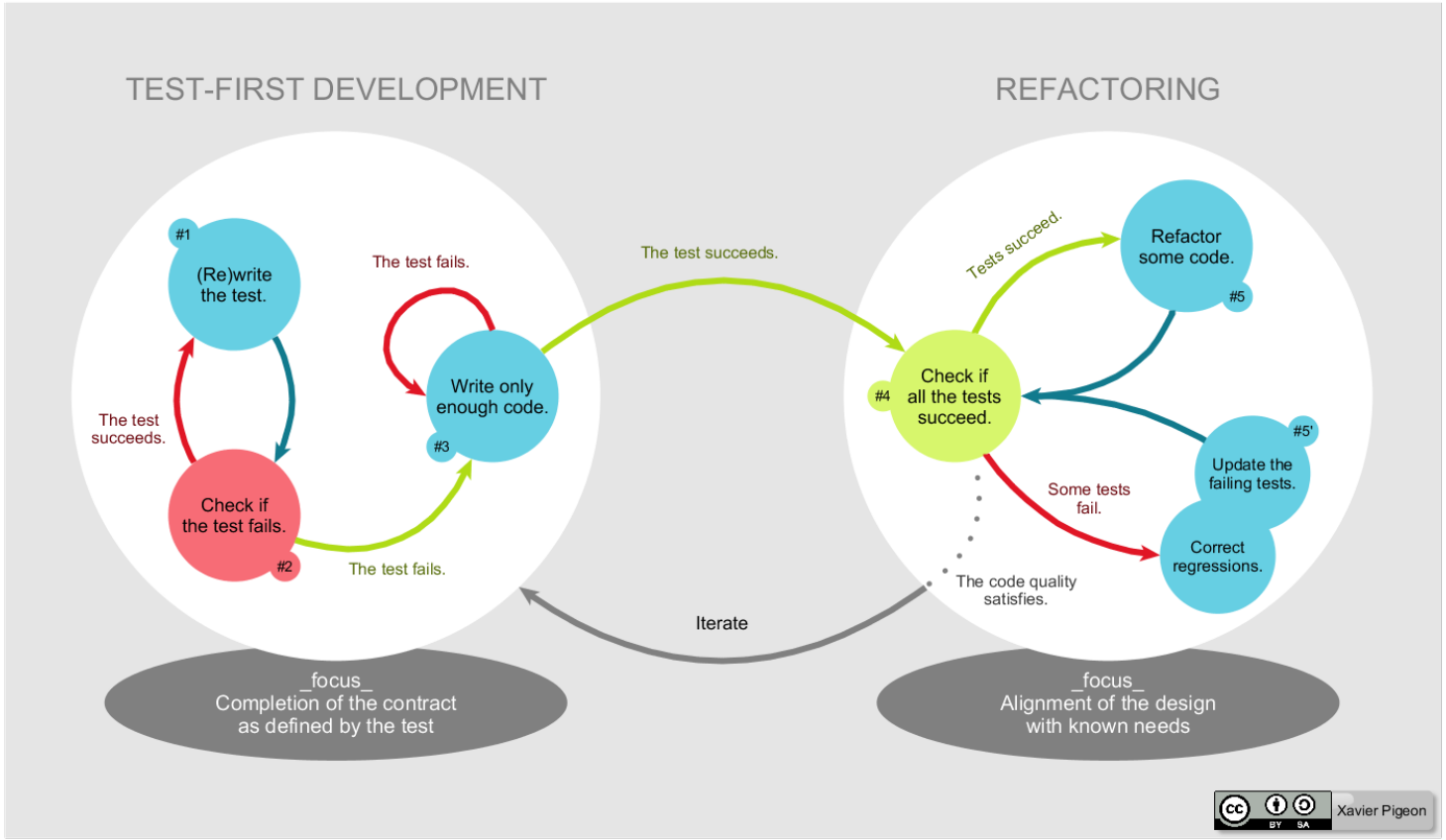
1. Figure out what the class is supposed to *do*.
2. List the **instance variables and methods**.
3. Write **prep code** for the methods.
4. Write **test code** for the methods.
5. **Implement** the class.
6. **Test** the methods.
7. **Debug** and **re-implement** as needed.

1. prep code
A form of ***pseudo-code***, to help you focus on the logic without stressing about syntax.

2. test code
A class or methods that will test the real code and validate that it's doing the right thing.

3. real code
The actual implementation of the class. This is where we write real Java code.

Test-driven development



[from Wikipedia]

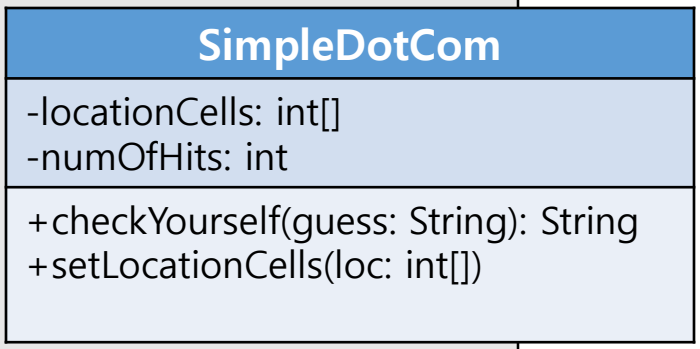
Prep Code

```
DECLARE an int array to hold the location cells. Call it locationCells.
DECLARE an int to hold the number of hits. Call it numOfHits and SET it to 0.
```

```
DECLARE a checkYourself() method that takes a String for the user's guess ("1", "3",
etc), checks it, and returns a result representing a "hit", "miss", or "kill".
DECLARE a setLocationCells() setter method that takes an int array (which has the three
cell locations as ints (2, 3, 4, etc.).
```

```
METHOD: String checkYourself(String userGuess)
   GET the user guess as a String parameter
   CONVERT the user guess to an int
   REPEAT with each of the location cells in the int array
      // COMPARE the user guess to the location cell
      IF the user guess matches
         INCREMENT the number of hits
         // FIND OUT if it was the last location cell:
         IF number of hits is 3, RETURN "kill" as the result
         ELSE it was not a kill, so RETURN "hit"
         END IF
      ELSE the user guess did not match, so RETURN "miss"
      END IF
   END REPEAT
END METHOD
```

```
METHOD: void setLocationCells(int[] cellLocations)
   GET the cell locations as an int array parameter
   ASSIGN the cell location parameter to the cell locations instance variable
END METHOD
```



Test Code (1)

- What we should test:
  1. Instantiate a *SimpleDotCom* object.
  2. Assign it a location (an array of 3 ints, like {2, 3, 4}).
  3. Create a *String* to represent a user guess ("2", "0", etc.).
  4. Invoke the *checkYourself()* method, passing it the fake user guess.
  5. Print out the result to see if it's correct ("passed" or "failed").

Test Code (2)

```
public class SimpleDotComTestDrive {
   public static void main (String [] args) {
      Instantiate a SimpleDotCom object.

      Assign it a location (an array of 3 ints, like {2, 3, 4}).

      Create a String to represent a user guess ("2", "0", etc.).

      Invoke the checkYourself() method, passing it the fake user guess.

      Print out the result to see if it's correct ("passed" or "failed").
   }
}
```

Real Code (1)

```
public class SimpleDotCom {
   private int [] locationCells;
   private int numOfHits = 0;

   public void setLocationCells(int [] locs) {
      if (locs != null && locs.length > 0) {
         this.locationCells = new Array[locs.length];
         System.arraycopy(locs, 0, this.locationCells, 0, locs.length);
      }
   }

   public String checkYourself(String stringGuess) { ... }
}
```

Why?

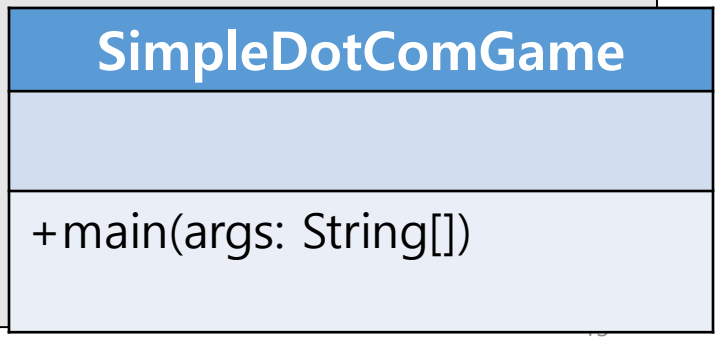
Real Code (2)

```
public class SimpleDotCom {
   private int [] locationCells;
   private int numOfHits = 0;

   public void setLocationCells(int [] locs) { ... }
   public String checkYourself(String stringGuess) {

      int guess = Integer.parseInt(stringGuess);
      String result = "miss";
      for (int cell : this.locationCells) {
         if (guess == cell) {
            result = "hit";
            this.numOfHits++;
            break;
         }
      }
      if (this.numOfHits >= this.locationCells.length) {
         result = "kill";
      }
      return result;
   }
}
```

SimpleDotComGame (Prep)

DECLARE an int variable to hold the number of user guesses, named ***numOfGuesses***, set it to 0.
MAKE a new SimpleDotCom instance.
COMPUTE a random number between 0 and 4 that will be the starting location cell position.
MAKE an int array with 3 int's using the randomly-generated number, that number incremented by 1, and that number incremented by 2 (example: 3, 4, 5)
INVOKE the ***setLocationCells()*** method on the SimpleDotCom instance
DECLARE a boolean variable representing the state of the game, named ***isAlive***. **SET** it to true

WHILE the dot com is still alive (isAlive == true):
  GET user input from the command line
  INVOKE the checkYourself() method on the SimpleDotCom instance
  INCREMENT numOfGuesses variable
  IF result is "kill"
    SET ***isAlive*** to false
    PRINT the number of user guesses
  END IF
END WHILE



SimpleDotComGame (Real)

```
public static void main(String [] args) {
    DECLARE an int variable to hold the number of user guesses, named numOfGuesses, set it to 0.
  GameHelper helper = new GameHelper();

    MAKE a new SimpleDotCom instance.

    COMPUTE a random number between 0 and 4 that will be the starting location cell position.
    MAKE an int array with 3 ints using the randomly-generated number, that number incremented by 1,
          and that number incremented by 2 (example: 3, 4, 5)
    INVOKE the setLocationCells() method on the SimpleDotCom instance.

    DECLARE a boolean variable representing the state of the game, named isAlive. SET it to true

  while (isAlive)  {
      GET user input from the command line.
      INVOKE the checkYourself() method on the SimpleDotCom instance.
      INCREMENT numOfGuesses variable.
      IF result is "kill"
          SET isAlive to false
          PRINT the number of user guesses
      END IF
  }
}
```

Random & getUserInput()

- Random

```
// package java.lang
// class Math
// static double random()
// returns a double value with a positive sign, greater than or
// equal to 0.0 and less than 1.0
int randomNum = (int)(Math.random() * 5);
```

- getUserInput()

```
String guess = helper.getUserInput("enter a number");
```

GameHelper (Real)

```
import java.io.*;

public class GameHelper {
  public String getUserInput(String prompt) {
    String inputLine = null;
    System.out.print(prompt + " ");
    try {
      BufferedReader is = new BufferedReader(new InputStreamReader(System.in));
      inputLine = is.readLine();
      if (inputLine.length() == 0) return null;
    } catch (IOException e) {
      System.out.println("IOException: " + e);
    }
    return inputLine;
  }
}
```

GameHelper (Real)

```
import java.io.*;

public class GameHelper {
  public String getUserInput(String prompt) {
    String inputLine = null;
    System.out.print(prompt + " ");
    try {
      BufferedReader is = new BufferedReader(new InputStreamReader(System.in));
      inputLine = is.readLine();
      if (inputLine.length() == 0) return null;
    } catch (IOException e) {
      System.out.println("IOException: " + e);
    }
    return inputLine;
  }
}
```

GameHelper (Real)

```
import java.io.*;

public class GameHelper {
  public String getUserInput(String prompt) {
    String inputLine = null;
    System.out.print(prompt + " ");

    Scanner input = new Scanner(System.in);
    inputLine = input.nextLine();




    return inputLine;
  }
}
```

class InputStreamReader (1)

```
// package java.io
// InputStreamReader
// InputStreamReader(InputStream in)
// or InputStreamReader(InputStream in, Charset cs)

InputStreamReader in = new InputStreamReader(System.in);

InputStreamReader in = new InputStreamReader(System.in,
                    java.nio.charset.StandardCharsets.UTF_8);

    // ISO_8859_1, US_ASCII, UTF_16, UTF_16BE, UTF_16LE, UTF_8
```

class Math

- Fields

```
static double E;    // the base of the natural logarithms.
static double PI;   // the ratio of the circumference of
                    // a circle to its diameter
```

```
// example
double area = radius * radius * Math.PI;
```

- Methods

```
static double abs(double a)
static float abs(float a)
static int abs(int a)
static long abs(long a)
…
// min, max, sin, cos, tan, toDegrees, toRadians, random, pow, ...
```

```
// example
double positive_value = Math.abs(-3.453);
```

Converting a String to an int

- Converting a String to an int

```
String stringGuess = "2";
int guess = Integer.parseInt(stringGuess);
```

- Converting a String to a double

```
String stringGuess = "2.2";
double guess = Double.parseDouble(stringGuess);
```

- Converting a double to a String

```
double v = 3.4;
v /= 1.23;
String strV = Double.toString(v);
```

Review: Loops

- *for, while, do-while*

```
while (x > 12) {
    x = x - 1;
}

do { x = x – 1; } while (x > 12);

double[] data = {1.0, 2.8, 3.2};
for (i = 0; i < data.length; i++) {
    System.out.println("data: " + data[i]);
}

for (double datum : data) {
    System.out.println("data: " + datum);
}
```

Casting primitives

- Casting primitives

```
long y = 42L;
int x = y;    // won't compile
int x = (int)y;
```

```
long y = 40002L;
short x = (short) y; // x now equals -25534!
```

```
float f = 3.14f;
int x = (int) f; // x will equal 3
```

- An example using a method such as Math.round()

```
float f = 3.59f;
// static int round(float a)
int x = Math.round(f); // x will equal 4
```

References

- Kathy Sierra and Bert Bates, *Head First Java*, O'Reilly, 2005.
- Battleship (Game), Wikipedia.
  - http://en.wikipedia.org/wiki/Battleship_(game)
- The Java Tutorials: Array
  - http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html

Q&A