Life Update - see second page for details

deep and shallow copies
counting neighbors

Anatomy of a Java class - CoinPurse / money

class definition
defines new class
class name should start with a capital letter

instance variables
hold state of the object
each instance gets its own copy of the instance variables
usually private
users must use the class’s methods to access or change the instance variables

class invariant
precise description of how the instance variables represent the object.
states what is true for a correctly formed instance of the class
each method can depend on that invariant being true when it starts
and must ensure that it is true when it finishes

constructors
method that initializes the instance variables
same name as the class
no return value because constructor is not a method
by default, the instance variables have default value for their type
can have a no argument constructor
constructor’s responsibility is to create a valid object
throw IllegalArgumentException if arguments cannot create valid object
that lets calling program decide what to do

methods
accessors and mutators
accessors - return existing information without changing class state
CoinPurse - value, numQuarters, numDimes, ..., toString, clone, equals
mutators - change the class state by changing the instance variables
have to be careful to maintain the class invariant
CoinPurse - addQuarters, ..., transferFrom
this is a special identifier referring to this object
can refer to other object’s instance variables with other.dimes
other is NOT a special name; could say wallet.dimes
If another instance of the class is passed as an argument,
method code can access its instance variables directly
4. (Feb. 1) For each cell in the active part of this board, write in the cell the number of live neighbors it has.

```
- - - - - -
- # - # - -
- # # # # -
- - - # # -
- # - # - -
- - - - - -
```

```
0 0 0 0 0
- - # - - 1
- # 6 # # 2
- - # # # 3
- # - # - 4
- - - - - 5
```

Write a function to calculate number of neighbors including this cell.
Cell - row index and col index
Board
```
public static int numNeighbors(boolean[][] board, int row, int col) {
    for (int rd = -1; rd <= 1; rd++) { // rd == row delta
        for (int cd = -1; cd <= 1; cd++) { // cd == column delta
            row + rd
            if (board[row+rd][col+cd])
                // make a deep copy
                boolean[][] original = theBoard.clone();
                for (int row = 0; row < theBoard.length; row++) {
                    original[row] = theBoard[row].clone();
                }
}
```

**Deep vs. Shallow Copy**

- M → N
  - M's values point to
  - N’s immediate values
- M → N
  - M's copy copies only
  - M's immediate values
- Deep copy makes a complete separate copy