

Refactoring In-Class Assignment

Due: End of class, 9/14

Do as much of this assignment as you can in the time remaining for class. You may work with one other person to do this assignment. Each pair should submit only one set of answers through the drop box on blackboard. **REMEMBER TO PUT YOUR NAMES IN THE FILE NAME AND INSIDE THE FILE.** If you have completed all the questions before the scheduled end of class, you may leave early.

1. Extract a telephone number class from the following person class:

```
class Person....
    public String getName() {
        return _name;
    }

    public String getTelephoneNumber() {
        return (“(“ + _officeAreaCode + “) “ + _officeNumber);
    }

    String getOfficeAreaCode() {
        return _officeAreaCode;
    }

    void setOfficeAreaCode(String arg) {
        _officeAreaCode = arg;
    }

    String getOfficeNumber() {
        return _officeNumber;
    }

    void setOfficeNumber(String arg) {
        _officeNumber = arg;
    }

    private String _name;
    private String _officeAreaCode;
    private String _officeNumber;
```

2. Replace the magic numbers in the following code with symbolic constants. Don't worry about changing the test code or doing other refactorings. **(BONUS: I'll give a bonus point to everyone who successfully performs other refactorings on this code. To get this point, you must include the refactoring(s) you performed, the reasons you chose each refactoring, and the code after EACH refactoring. Giving me one block of code and a list of more than one refactoring applied to it does not get the point. The bonus portion of this question can be submitted after the end of class)**

```
import junit.framework.*;

public class GameTest extends TestCase {
    public GameTest(String s) {super(s);}

    public void testDefaultMove() {
        Game game = new Game("XOXOX-OXO");
        assertEquals(5, game.move('X'));

        game = new Game("XOXOXOOX-");
        assertEquals(8, game.move('O'));

        game = new Game("-----");
        assertEquals(0, game.move('X'));

        game = new Game("XXXXXXXXXX");
        assertEquals(-1, game.move('X'));
    }

    public void testFindWinningMove() {
        Game game = new Game("XO-XX-OOX");
        assertEquals(5, game.move('X'));
    }

    public void testWinConditions() {
        Game game = new Game("---XXX---");
        assertEquals('X', game.winner());
    }
}

public class Game {
    public StringBuffer board;

    public Game(String s) {board = new StringBuffer(s);}

    public Game(StringBuffer s, int position, char player) {
        board = new StringBuffer();
    }
}
```

```

        board.append(s);
        board.setCharAt(position, player);
    }

    public int move(char player) {
        for (int i = 0; i < 9; i++) {
            if (board.charAt(i) == '-') {
                // Create a temporary game to determine
                // if a given move is a winning move
                Game game = play(i, player);
                if (game.winner() == player)
                    return i;
            }
        }

        for (int i = 0; i < 9; i++) {
            if (board.charAt(i) == '-')
                return i;
        }

        return -1;
    }

    public Game play(int i, char player) {
        return new Game(this.board, i, player);
    }

    public char winner() {
        if (board.charAt(0) != '-')
            && board.charAt(0) == board.charAt(1)
            && board.charAt(1) == board.charAt(2))
            return board.charAt(0);
        if (board.charAt(3) != '-')
            && board.charAt(3) == board.charAt(4)
            && board.charAt(4) == board.charAt(5))
            return board.charAt(3);
        if (board.charAt(6) != '-')
            && board.charAt(6) == board.charAt(7)
            && board.charAt(7) == board.charAt(8))
            return board.charAt(6);
        return ('-');
    }
}

```

3. Simplify the following conditionals:

a. `if (!_currentMonth != 2)
 if (((_currentYear mod 4 == 0) && (_currentYear mod 100 != 0))
 || (_currentYear mod 400 == 0)) {
 //it's a leap year`

b. `if (((elementNo < 0) || (elementNo > maxElement)) ||
 (elementNo == lastElement)) {
 :
}`

c. `if (!(_currentlyEnrolled && _hasProfessorsRespect) &&
 !(_graduateStudent || _childOfRichAlumnus)
 fail();`