

Lucas Cowden

14 April 14, 2022

M5 Code Smells

```
public abstract class Tile {  
    private int x;  
    private int y;  
    public static final int WIDTH = 30;  
    public static final int HEIGHT = 30;  
  
    public abstract void draw(int x, int y, GraphicsContext gc, int layer);  
}
```

```
public abstract class AbstractTower extends Tile {
```

```
public class BCTower extends AbstractTower {  
    private ArrayList<Path> adjacentPaths;  
    private Wall wall;  
    private Map map;  
    private int x;  
    private int y;
```

There is an instance of duplicated code in the above pictures where the Tile class has an x and a y variable declared, but a grandchild class of Tile also has an x and a y variable declared. This means we are redeclaring what should be the same object, which is both sloppy and inefficient. Storage space could be excessively used in this scenario, and we should seek to only use the Tile's instance of x and y as every Tile child class can utilize this data.

```
public void setX(int x) {  
    ((Tile) this).setXVar(x);  
}  
  
public void setY(int y) {  
    ((Tile) this).setYVar(y);  
}
```

```
if (this.map.getTileInDirection(this.map.getCoords(((Tile) this).getX(),
    ((Tile) this).getY()), dir) instanceof Path) {
    adjacentPaths.add((Path) this.map.getTileInDirection(this.map.getCoords(((Tile) this).getX(),
        ((Tile) this).getY()), dir));
}
```



The screenshot shows an IDE with four tabs: Grass.java, Player.java, Map.java, and Tile.java. The Tile.java tab is active, displaying the following code:

```
public void setXVar(int x) {
    this.x = x;
}

public void setYVar(int y) {
    this.y = y;
}

public int getX() {
    return this.x;
}

public int getY() {
    return this.y;
}
```

To fix this I wrote getter and setter methods for x and y, and during calls to the x and y in question I utilize the getX() and getY() functions. When initializing a BCTower object, I use the setter methods I wrote in the Tile class to set the x and y values. Furthermore I removed the local variables x and y in BCTower, making our code more efficient and keeping in line with the principles of inheritance.