Универзитет Св.Кирил и Методиј
Факултет за Информатички науки и Компјутерско Инженерство

Дизајн и архитектура на софтвер

# Проект за рефакторирање

Автор: Стефан Цветковски
Број на индекс: 111241, е-пошта: cvetkovski.stefan@students.finki.ukim.mk

## Package Battleships_____

- **Agent.java**
  - Поправен конструкторот. Наместо public void Agent( ) => public Agent( )
- **BattleShipsEngine.java**
  - Употребен **Образец метод шаблонот**, **Extract & Move Method, Add Class to Hierarchy**
    - Во оваа класа се наоѓа и главниот main метод, но и целата логика на играта. Целиот код за логиката на играта беше напишан структурно во самиот main метод. Ова не е практично, затоа што јас понатака ќе сакам main методот да го префрлам во посебна класа која ќе ми означува и стартер, и сегашниов код нема да работи. Од таа цел, го употребив Образец метод за да креирам метод start во кој ќе биде дефиниран скелет на алгоритамот и фиксни чекори.

```
public abstract class BattleshipEngineTemplate {
    public final void start() {
        init();
        deployShips();
        setPlayerTurn();
        playGame();
        gameOver();
    }

    public abstract void init();
    public abstract void deployShips();
    public abstract void setPlayerTurn();
    public abstract void playGame();
    public abstract void gameOver();
}
```

Секако со ова се креира и нова класа BattleshipEngineTepmlate.java.

- Сега комплетно можам main методот да го префрлам во посебна класа која ќе ми биде стартер (основна, play, run класа)

```
public static void main (String args[])
{
    BattleShipsEngine engine = new BattleShipsEngine();
    engine.start();
}
```

  - Употребен **Extract Method**
    - Првично методата private static void determineIfShotSunkAShip(GUI gui, Agent smith), е променета во private void determineIfShotSunkAShip( ). Нема потреба да биде static поради користење на Образец методот од претходно, и нема потреба да прима аргументи затоа што се однесува на променливи од класата.
    - Следно, во оваа класа се наоѓаа 5 услови, независни еден од друг за проверка дали е погоден објект или дали е уништен, и притоа се проверуваше за сите 5 објекти (Submarine, Destroyer, Battleship, AircraftCarrier, Minesweeper). Секој од тие услови беше изваден и поставен во свој посебен метод. (Овој чекор има повторно налик како Образец методот)

```
private void determineIfShotSunkAShip() {
    System.out.println("Player Home board \n" +gui.data.gameState.playerHomeGrid.toString());
    isMineSunk();
    isDestSunk();
    isSubSunk();
    isBattleSunk();
    isAirSunk();
}
private void isMineSunk(){

private void isDestSunk(){

private void isSubSunk(){

private void isBattleSunk(){

private void isAirSunk(){
```

  - Употребен **Move Method**
    - Префрлање на main методот од BattleShipsEngine.java во Play.java која се наоѓа во паќетот Battleships.main
    - Со ова се добива многу јасна претстава каде е извршната класа и пртиоа во извршната класа го имаме само извршувањето.

```
package Battleships.main;

import Battleships.BattleShipsEngine;

public class Play {
    public static void main (String args[])
    {
        BattleShipsEngine engine = new BattleShipsEngine();
        engine.start();
    }

}
```

- **GameState.java**
  - Употребено **Remove Variable**

        - Оваа класа беше солидно напишана и многу лесна за разбирање. Сепак во целиот код на класава променливата isDeployedComplete е само декларирана, но никаде не е иницијализирана и употребена. Затоа оваа променлива беше отстранета.
- **Grid.java**
    - Тргнат import java.util.concurrent.ExecutionException од причина што не се користи
    - **@SuppressWarnings("serial")** – додадена нотација на класата
    - Отстранет закоментиран код
    - Променет Метод
        - Методот public boolean allShipsSunk( ), е променет со цел да се искористат неколку getter методи, одколку самите приватни променливи.

```
public boolean allShipsSunk() {
    return checkAirSunk() && checkBattleSunk() && checkDestSunk()
            && checkSubSunk() && checkMineSunk();
}
```

        - Со оваа измена, се доби една непотребна променлива која ќе биде отстранета
        - Во методот public boolean allShipsPlaced( ), избришани се условите и телото на методот е намалено и поедноставено. Одма добиваме одговор дали сите бродови се поставени со помош на следниве помошни методи.

```
/** Checks if all ships have been placed */
public boolean allShipsPlaced() {
    return checkMinePlaced() && checkSubPlaced() && checkDestPlaced()
            && checkBattlePlaced() && checkAirPlaced();
}
```

        - Слична постапка беше спроведена на методите *checkMinePlaced*, *checkSubPlaced*, *checkAirPlaced*.
    - Употребено **Remove Variable**
        - Од кога се направи измена на методот опишан погоре, доведе да променливата *boolean allShipsSunk* не се користи повеќе нигдека и е отстранета.
        - Во методот public boolean shot(int i, int j) локалната променлива *String output* прима вредност но нигдека не се печати, ниту користи. Претпоставувам дека било потребно за дебагирање, но сега немаме корист од неа. Променливата е отстранета.
    - Употребено **Remove Method**
        - Методот *public String printIsPlaced( )* не се применува и е отстранет.
        - Методот *public String printIsSunk ( )* не се применува и е отстранет.
- **GUI.java**
    - **@SuppressWarnings("serial")** – додадена нотација на класата
    - Тргнати следниве import-и:
        - **import java.awt.event.*;**
        - **import javax.swing.border.*;**
        - **import java.util.*;**
    - Употребено **Add Class to Hierarchy**, **Move Method**
        - Во класата GUI, во долните редови од дадотеката има дефинирано 6 Listener-и, и притоа тие Listener-и се класи. Во класата имаме дефинирано уште 6 класи. За секоја од овие класи креирав соодветна дадотека .java и ги префрлив таму.
        - AttackMousePressListener, `HomeMousePressListener, HideButtonAction, RotateButtonAction, QuitButtonAction, ShowButtonAction`

- Употребено **Remove Method**
  - Методот public void paintComponent(Graphics g) е отстранет поради тоа што не се применува никаде, а притоа методот нема ни тело. Празна функција.
- Употребено **Remove Variable**
  - Променливите Grid compHome и Grid compAtt се иницијализирани локално но не се применуваат. Од таа причина се отстранети.
  - Пред да се префрлат Listener-ите во посебни дадотеки, во нив беше отсратена променливата Graphics g, затоа што таа не се употребуваше.

- **InfluenceMap.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Употребено **Remove Method**
    - Прво приметив дека променливата int Dave во методот public void setDeadends(int i, int j) нема никакво значение, исто и boolean done. Подоцна приметив дека и целиот метод уствари не прави ништо, ниту се повикува од некоја друга класа. Приметив дека методот public void searchDeadends( ) го повикува претходниот метод, но како што реков нема никаква цел, а ни овој метод исто така не е повикан од ниеден друг метод ниту класа. Прво го закоментирав кодот и ја тестирав целата апликација и видов дека не менува потполно ништо. Одлучив да ги тргнам тие 2 методи.
  - Употребено **Extract Method**
    - Методата public void sunk(int i, int j) имаше премногу големи услови доколку се случи погодок, па кде погодокот е случен со цел да се предвиди каде би можел да биде пловниот објект за да се уништи. Вкупно имаше 5 главни услови кои ги извадив во посебни методи. Новиот код што се доби за методот sunk излгеда вака:

```
public void sunk(int i, int j) {
    if (map[i][j] == hit) {
        // if the hit is not on an edge
        hitIsNotOnAnEdge(i, j);
        // if hit is on left collumn but not top left or bottom lef
        hitIsOnLeftCollumnButNotTopOrBottomLeft(i, j);
        // if hit is on right most column but not top right or bottom right
        hitIsOnRightCollumnButNotTopOrBottomRight(i, j);
        // if hit on bottom row
        hitOnBottomRow(i, j);
        // if hit is on top row
        hitOnTopRow(i, j);
    }
}
```

    - Слична ситуација беше и со методата public void hit(int i, int j), но наместо услови имавме 4 try-catch блокови. Бидејќи секој од тие блокови беше независен, беше изваден во посебен метод.

```
public void hit(int i, int j) {
    map[i][j] = hit;

    southernIsNotAHit(i, j);
    nothernIsNotAHit(i, j);
    easternIsNotAHit(i, j);
    westernIsNotAHit(i, j);
}
```

  - Употребено Add Method, бришење на код што се дуплира
    - Приметив дека методите getHotspotJ() и getHotspotI() имаат наполно идентичен код, единствена разлика беше во return вредноста на методот, па одлучив да

креирам нов метод public int getHotSpot(char a). Притоа аргументот a е контролен параметар и во зависност од него се враќа вредност на j(x) или i(y).

```
public int getHotSpot(char a) {
    int x = 0;
    int y = 0;
    int val = 0;

    for (int i = 0; i < 10; i++)
        for (int j = 0; j < 10; j++) {

            if (map[i][j] > val && map[i][j] != hit) {
                val = map[i][j];
                y = i;
                x = j;
            }
        }
    if (a == 'j')
        return x;
    if (a == 'i')
        return y;

    return 0; // Vo slucaj da vnese pogreshna bukva
}
```

  - Употребено **Remove Variable**
    - Отстранета е променливата private int HotSpots = 0; од причина што не се применува нигдека.

- **MapStore.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Додаден само Generic type на ArrayList → ArrayList<InfluenceMap>
  - Нема промени во кодот
- **NumberGenerator.java**
  - Нема промени во кодот

## Package Battleships.exception_____

- **@ SuppressWarnings("serial")**
  - Додадена нотација на сите класи во овој пакет
- **IncorrectOrientationParameterException.java**
  - Нема промени во кодот
- **InitialPositionOccupiedException.java**
  - Нема промени во кодот
- **OutOfGridAreaException.java**
  - Нема промени во кодот
- **PositionExceedsBoardException.java**
  - Нема промени во кодот
- **PositionOccupiedException.java**
  - Нема промени во кодот

## Package Battleships.Graphics_____

- **AircraftCarrier.java**
  - Отстранет закоментиран код
  - Употребено **Extract Method**, **Move Method** и **Remove Class Method**
    - Во AircraftCarrier класата се наоѓа код за исцртување на вертикален брод. Притоа има друга класа AircraftCarrierH за исцртување на истиот брод но во хоризонтална положба. Наполно овие 2 класи се потполно исти и кодот се повторува, па затоа решив да ги спојам во една со тоа што го извадив кодот од AircraftCarrierH за цртање на хоризонтален брод и го додадов во новиот метод paintHorizontal во AircraftCarrier. Исто и во AircraftCarrier го извадив кодот за цртање вертикален брод и го додадов во нов метод paintVertical.
    - Следно беше потребно како да знаеме каков брод да нацртаме, па затоа во методот paint додадов контролен аргумент direction кој ни помага каков прод да нацртаме.

```
public static void paint(Graphics g, int xLeft, int yTop, char direction)
{
            if(direction == 'v') paintVertical(g, xLeft, yTop);
            if(direction == 'h') paintHorizontal(g, xLeft, yTop);
}
```

    - Класата AircraftCarrierH.java беше отстранета.
    - ****Сосема слична (иста) постапка имаа и класите Battleship и BattleshipH, Destroyer и DestroyerH, Minesweeper и MinesweeperH, Submarine и SubmarineH. Да не се повторувам во секоја од нив, смао ќе се повикам на овие промени****
- **AircraftCarrierH.java**
  - Употребено **Move Method** и **Remove Class Method**
    - Објаснето погоре во **AircraftCarrier.java**
- **AttackPanel.java**
  - Тргнат import jawa.awt.event.* од причина што не се користи
  - **@SuppressWarnings("serial")** – додадена нотација на класата
- **Battleship.java**
  - Отстранет закоментиран код
  - Употребено **Extract Method**, **Move Method** и **Remove Class Method** ***
    - Сосема потполно исто како и во **AircraftCarrier.java**, но сега применето на класите Battleship и BattleshipH
- **BattleshipH.java**
  - Употребено **Move Method** и **Remove Class Method** ***
- **Destroyer.java**
  - Отстранет закоментиран код
  - Употребено **Extract Method**, **Move Method** и **Remove Class Method** ***
- **DestroyerH.java**
  - Употребено **Move Method** и **Remove Class Method** ***
- **DrawGrid.java**
  - Отстранет закоментиран код
- **GridPanel.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Нема променив во кодот
- **HitIcon.java**
  - Отстранет закоментиран код
- **HomePanel.java**

- **@SuppressWarnings("serial")** – додадена нотација на класата
- Отстранет закоментиран код

- **InfluenceMapGraphic.java**
  - Отстранет закоментиран код
  - Употребено **Extract Method (extract code in new method)**
    - Методот paint се состоеше од 2 дела: избирање на боја и исцртување на форма. Бидејќи избирањето на боја зависеше од одреена вредност, во овој дел се наоѓаа над 10тина if услови за тоа која боја ќе се одбере. Па овој дел го извадив и го ставив во посебен метод *setColor(Graphics g, int val)*.

```
public static void paint(Graphics g, int xLeft, int yTop, int val) {

    setColor(g, val);

    g.fillRect(xLeft, yTop, 20, 20);

}
```

    - Бидејќи поставувањето на боја во *setColor* зависи од вредноста val променливата, во секој од условите е додаден return ; со цел да не се проверува во сите услови за тоа која боја да се земе. Условите се независни и затоа не може да се случи за иства вредноста на val да има две различни бои.
  - Употребено **Remove Variable**
    - Променливата Color mapVal не е потребна, од причина што нема некоја значителна функција. Заменета е со new Color(…) во неколку делови од кодот.
- **InfluencePanel.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Отстранет закоментиран код
  - Употребено **Remove Variable**
    - Променливата *InfuenceMap m* е само декларирана, но никаде не се иницијализира ниту применува. Од таа причина е отстранета.
  - Тргнат import *Battleships.InfluenceMap* – од причина што не се користи
- **Map.java**
  - Употребено **Remove Class**
    - Оваа класе е целосно отстранета од причина што не се користи ниту повикува од ниту една друга класа. Не влијае на извршувањето на апликацијата.
- **Minesweeper.java**
  - Отстранет закоментиран код
  - Употребено **Extract Method**, **Move Method** и **Remove Class Method** ***
    - Сосема потполно исто како и во **AircraftCarrier.java**, но сега применето на класите Minesweeper и MinesweeperH
- **MinesweeperH.java**
  - Употребено **Move Method** и **Remove Class Method** ***
- **MissIcon.java**
  - Отстранет закоментиран код
- **SplashIcon.java**
  - Употребено **Remove Class**
    - Оваа класа е потполна копија на MissIcon класата, целосно дуплиран код но само сменето името на класата. Притоа оваа класа не се употребува нигдека и затоа е отстранета
- **Submarine.java**
  - Отстранет закоментиран код
  - Употребено **Extract Method**, **Move Method** и **Remove Class Method** ***

- **SubmarineH.java**
  - Употребено **Move Method** и **Remove Class Method** ***
- **SunkIcon.java**
  - Нема променив во кодот

## Package Battleships.Ships_____

- **AircraftCarrier.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Нема променив во кодот
- **Battleship.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Тргнати следниве import-и од причина што не се користат
    - **import Battleships.exception.PositionExceedsBoardException;**
    - **import Battleships.exception.PositionOccupiedException;**
  - Нема променив во кодот
- **Destroyer.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Нема променив во кодот
- **Minesweeper.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Нема променив во кодот
- **Ship.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Нема променив во кодот
- **Submarine.java**
  - **@SuppressWarnings("serial")** – додадена нотација на класата
  - Нема променив во кодот

## Package Battleships.main_____

- **Play.java**
  - Употребено **Add Class to Hierarchy**
    - Ова е нова креирана класа која кого содржи извршниот main метод.
    - Оваа класа и нејзината стуктура ја опишав уште погорев во BattleShipEngine

```
package Battleships.main;

import Battleships.BattleShipsEngine;

public class Play {
    public static void main (String args[])
    {
        BattleShipsEngine engine = new BattleShipsEngine();
        engine.start();
    }

}
```

## Package Battleships.patterns_____

- **BattleshipEngineTemplate.java**
    - Употребено **Add Class to Hierarchy , Образец метод шаблонот**
        - Како што реков претходно,  во оваа класа е сместен образец методот, кој го имплементира BattleShipsEngine.java. Повеќе за ова објаснив кога ја рефакторирав BattleShipsEngine класата.