



**UMS**  
UNIVERSITI MALAYSIA SABAH

# **KK14203**

## **OBJECT ORIENTED PROGRAMMING**

**REPORT**  
**PROJECT 2 (INDIVIDUAL)**

**LECTURER: DR. SAMRY@MOHD SHAMRIE SAININ**

<b>Name</b>	<b>Matrics No</b>
Jacintha Lucia Rumpungan	BI19110109

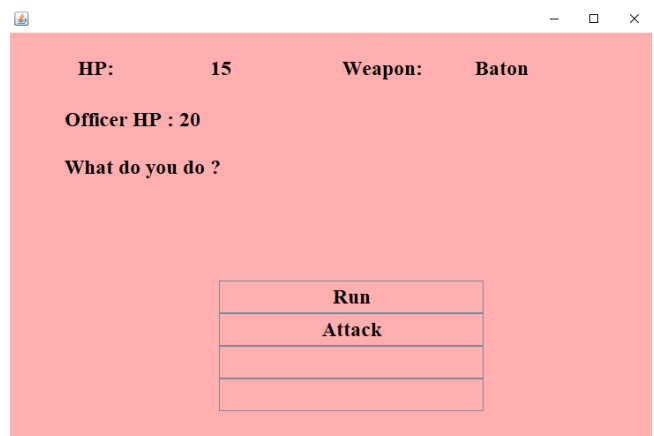
## GUI DESIGNS



Front page (Interface)

- This page shows the name of the game.
- There is a button, when user clicked it, the game will start.

- This is some of the story page

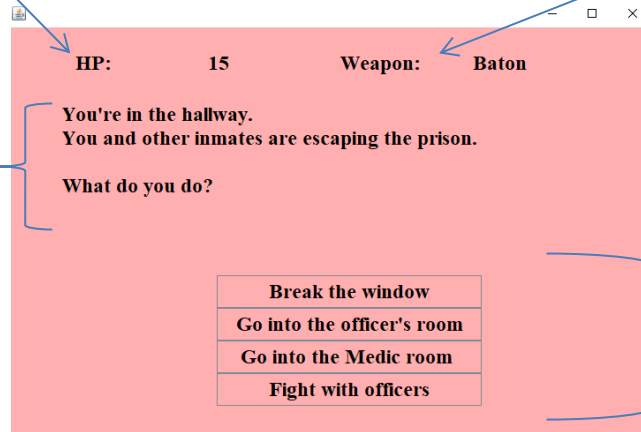


## USER MANUAL

This is where player's HP is shown.

This is where player's current weapon is shown. When player obtain a new weapon, the weapon will change name.

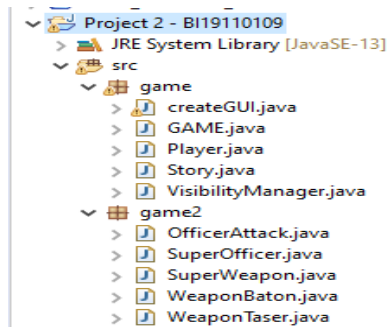
This is where the story or dialogues are shown.



This is where player can decide how the story goes, by choosing 1 of these 4 buttons. Each button will bring player to different places.

## CODING

*In this project I decided to use packages and do multiples class separately so other people can see the coding clearly.*



There are 2 packages, which are game and game2.

Package = game

class createGUI

```
Java Eclipse - Project 2 - BI19110109/src/game/createGUI.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

package game;
import java.awt.Color; //package for color

public class createGUI {
    JFrame window; //for window
    Container con;
    JPanel titleNamePanel, journeyButtonPanel, mainTextPanel, choiceButtonPanel, playerInfoPanel;
    JLabel titleLabel, titleLabel2, hpLabel, hpLabelNum, weaponLabel, weaponLabelName;
    Font titleFont = new Font("Times New Roman", Font.CENTER_BASELINE, 70); // set the font styles and font size
    Font titleFont2 = new Font("Times New Roman", Font.CENTER_BASELINE, 30);
    Font titleFont3 = new Font("Times New Roman", Font.CENTER_BASELINE, 23);
    JButton startButton, choice1, choice2, choice3, choice4;
    JTextArea mainTextArea;

    public void createGUI(ChoiceHandler cHandler) {
        //FOR WINDOW
        window = new JFrame(); //initialization
        window.setSize(750, 500); //width,height
        window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // so the window can close "X" button yg merah 2
        window.getContentPane().setBackground(Color.PINK);
        window.setLayout(null); // using no layout managers or can customize the layout
        con = window.getContentPane();

        //FOR TITLE SCREEN
        titleNamePanel = new JPanel();
        titleNamePanel.setBounds(70, 80, 600, 200);
        titleNamePanel.setBackground(Color.PINK);
        titleLabel = new JLabel("PRISON BREAK");
        titleLabel2 = new JLabel("- CHOOSE YOUR PATH -");
        titleLabel2.setForeground(Color.BLACK); // set the font color
        titleLabel2.setFont(titleFont); // call the title font
        titleLabel2.setFont(titleFont2);

        journeyButtonPanel = new JPanel();
        journeyButtonPanel.setBounds(250, 300, 200, 60);
        journeyButtonPanel.setBackground(Color.PINK);

        startButton = new JButton("LET'S GO");

        titleNamePanel.add(titleLabel);
        titleNamePanel.add(titleLabel2);
        journeyButtonPanel.add(startButton);

        con.add(titleNamePanel); // add title name in the container
        con.add(journeyButtonPanel); // add the button panel

        //FOR GAME SCREEN
        titleNamePanel.setVisible(false);
        journeyButtonPanel.setVisible(false);

        mainTextPanel = new JPanel();
        mainTextPanel.setBounds(40, 80, 550, 200);
        mainTextPanel.setBackground(Color.PINK);
        con.add(mainTextPanel);

        mainTextArea = new JTextArea("these is the main test area");
        mainTextArea.setBounds(40, 80, 550, 200);
        mainTextArea.setBackground(Color.PINK);
        mainTextArea.setForeground(Color.BLACK);
        mainTextArea.setFont(titleFont3);
        mainTextArea.setLineWrap(true); // convenient function to have( automatically lock text if write too much)
        mainTextPanel.add(mainTextArea);

        choiceButtonPanel = new JPanel();
        choiceButtonPanel.setBounds(240, 280, 300, 150);
        choiceButtonPanel.setBackground(Color.PINK);
        choiceButtonPanel.setLayout(new GridLayout(4,1)); // to set the choice buttons horizontally
        con.add(choiceButtonPanel);

        choice1 = new JButton("CHOICE 1");
        choice1.setBackground(Color.PINK);
        choice1.setForeground(Color.BLACK);
    }
}
```

```
Java Eclipse - Project 2 - BI19110109/src/game/createGUI.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

startButton = new JButton("LET'S GO");
startButton.setBackground(Color.PINK);
startButton.setBounds(40, 80, 550, 200);
startButton.setFont(titleFont3);
startButton.addActionListener(cHandler); // if we click the start button, than this function will react = the TitleScreenHandler will activated
startButton.setActionCommand("start");
startButton.setFocusPainted(false); // supaya tidak lagi nampak line time click button

titleNamePanel.add(titleLabel);
titleNamePanel.add(titleLabel2);
journeyButtonPanel.add(startButton);

con.add(titleNamePanel); // add title name in the container
con.add(journeyButtonPanel); // add the button panel

//FOR GAME SCREEN
titleNamePanel.setVisible(false);
journeyButtonPanel.setVisible(false);

mainTextPanel = new JPanel();
mainTextPanel.setBounds(40, 80, 550, 200);
mainTextPanel.setBackground(Color.PINK);
con.add(mainTextPanel);

mainTextArea = new JTextArea("these is the main test area");
mainTextArea.setBounds(40, 80, 550, 200);
mainTextArea.setBackground(Color.PINK);
mainTextArea.setForeground(Color.BLACK);
mainTextArea.setFont(titleFont3);
mainTextArea.setLineWrap(true); // convenient function to have( automatically lock text if write too much)
mainTextPanel.add(mainTextArea);

choiceButtonPanel = new JPanel();
choiceButtonPanel.setBounds(240, 280, 300, 150);
choiceButtonPanel.setBackground(Color.PINK);
choiceButtonPanel.setLayout(new GridLayout(4,1)); // to set the choice buttons horizontally
con.add(choiceButtonPanel);

choice1 = new JButton("CHOICE 1");
choice1.setBackground(Color.PINK);
choice1.setForeground(Color.BLACK);
}
```

```

Java Eclipse - Project 2 - B19110109/src/game/createGUI.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

createGUI.java
71
92 choice1 = new JButton("CHOICE 1");
93 choice1.setBackground(Color.PINK);
94 choice1.setForeground(Color.BLACK);
95 choice1.setFont(titleFont3);
96 choice1.setFocusPainted(false);
97 choice1.addActionListener(cHandler);
98 choice1.setActionCommand("c1");//to differentiate between the title button and these 4 choice buttons
99 choiceButtonPanel.add(choice1);
100
101 choice2 = new JButton("CHOICE 2");
102 choice2.setBackground(Color.PINK);
103 choice2.setForeground(Color.BLACK);
104 choice2.setFont(titleFont3);
105 choice2.setFocusPainted(false);
106 choice2.addActionListener(cHandler);
107 choice2.setActionCommand("c2");//to differentiate between the title button and these 4 choice buttons
108 choiceButtonPanel.add(choice2);
109
110 choice3 = new JButton("CHOICE 3");
111 choice3.setBackground(Color.PINK);
112 choice3.setForeground(Color.BLACK);
113 choice3.setFont(titleFont3);
114 choice3.setFocusPainted(false);
115 choice3.addActionListener(cHandler);
116 choice3.setActionCommand("c3");//to differentiate between the title button and these 4 choice buttons
117 choiceButtonPanel.add(choice3);
118
119 choice4 = new JButton("CHOICE 4");
120 choice4.setBackground(Color.PINK);
121 choice4.setForeground(Color.BLACK);
122 choice4.setFont(titleFont3);
123 choice4.setFocusPainted(false);
124 choice4.addActionListener(cHandler);
125 choice4.setActionCommand("c4");//to differentiate between the title button and these 4 choice buttons
126 choiceButtonPanel.add(choice4);
127
128
129 playerInfoPanel = new JPanel();
130 playerInfoPanel.setBounds(80, 15, 600, 50);
131 playerInfoPanel.setBackground(Color.PINK);
132 playerInfoPanel.setLayout(new GridLayout(1,4)); //custom layout
133 con.add(playerInfoPanel);
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159

```

```

133
134 hpLabel = new JLabel("HP: ");
135 hpLabel.setFont(titleFont3);
136 hpLabel.setForeground(Color.BLACK);
137 playerInfoPanel.add(hpLabel);
138
139 hpLabelNum = new JLabel();
140 hpLabelNum.setFont(titleFont3);
141 hpLabelNum.setForeground(Color.BLACK);
142 playerInfoPanel.add(hpLabelNum);
143
144 weaponLabel = new JLabel("Weapon: ");
145 weaponLabel.setFont(titleFont3);
146 weaponLabel.setForeground(Color.BLACK);
147 playerInfoPanel.add(weaponLabel);
148
149 weaponLabelName = new JLabel();
150 weaponLabelName.setFont(titleFont3);
151 weaponLabelName.setForeground(Color.BLACK);
152 playerInfoPanel.add(weaponLabelName);
153
154
155
156
157
158
159

```

## class GAME

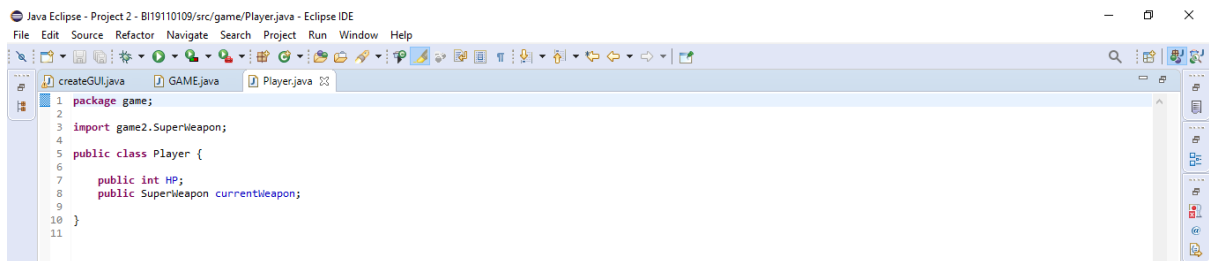
```

Java Eclipse - Project 2 - B19110109/src/game/GAME.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

createGUI.java GAME.java
1 package game;
2
3 import java.awt.event.ActionEvent;
4
5 public class GAME {
6
7     ChoiceHandler cHandler = new ChoiceHandler();
8     createGUI ui = new createGUI();
9     VisibilityManager vm = new VisibilityManager(ui);//send this to class VisibilityManager
10    Story story = new Story(this, ui, vm);
11
12    String nextPosition1, nextPosition2, nextPosition3, nextPosition4;
13
14    public static void main(String[] args) {
15        new GAME();
16    }
17
18    public GAME() {
19
20        ui.createGUI(cHandler);
21        story.GAMESetup();//to call this GAMESetup
22        vm.showTitleScreen();//to display title screen from class showTitleScreen
23    }
24
25    public class ChoiceHandler implements ActionListener{
26
27        public void actionPerformed(ActionEvent event) {
28
29            String yourChoice = event.getActionCommand();
30
31            switch (yourChoice) {
32                case "start" : vm.prison(); story.hallway(); break;
33                case "c1" : story.selectPosition(nextPosition1);break;
34                case "c2" : story.selectPosition(nextPosition2);break;
35                case "c3" : story.selectPosition(nextPosition3);break;
36                case "c4" : story.selectPosition(nextPosition4);break;
37            }
38        }
39    }
40
41
42

```

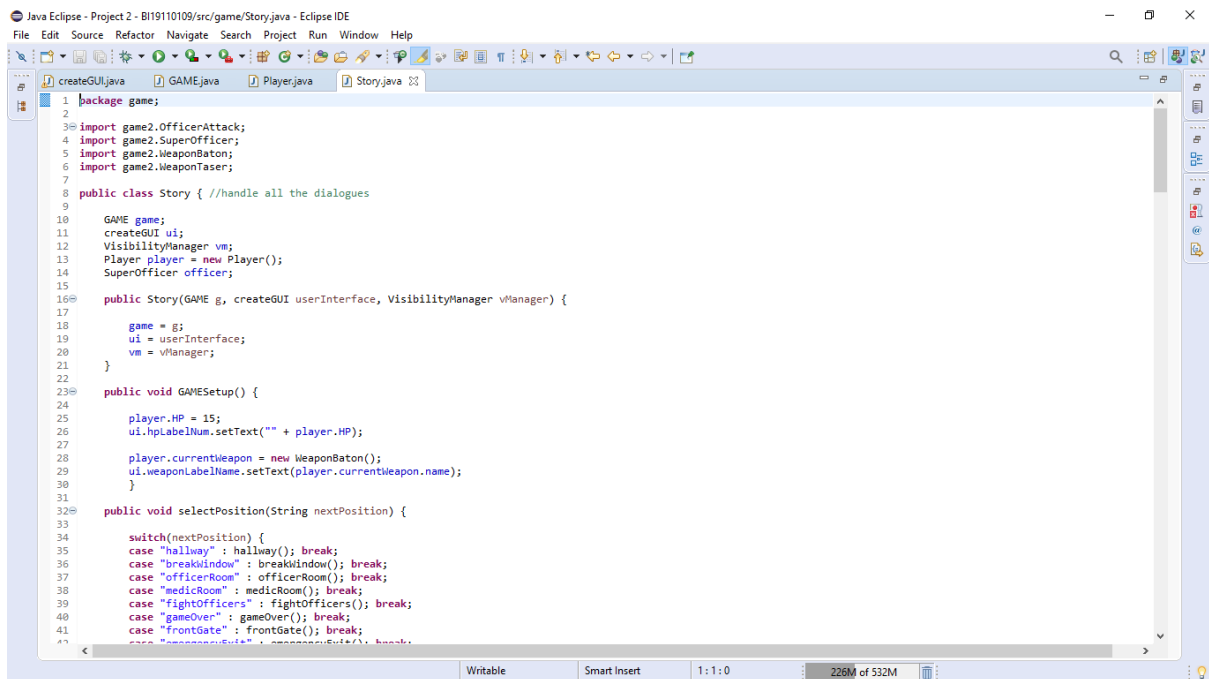
## class Player



The screenshot shows the Eclipse IDE with the 'Player.java' file open. The code defines a package 'game', imports 'game2.SuperWeapon', and defines a public class 'Player' with attributes 'HP' and 'currentWeapon'.

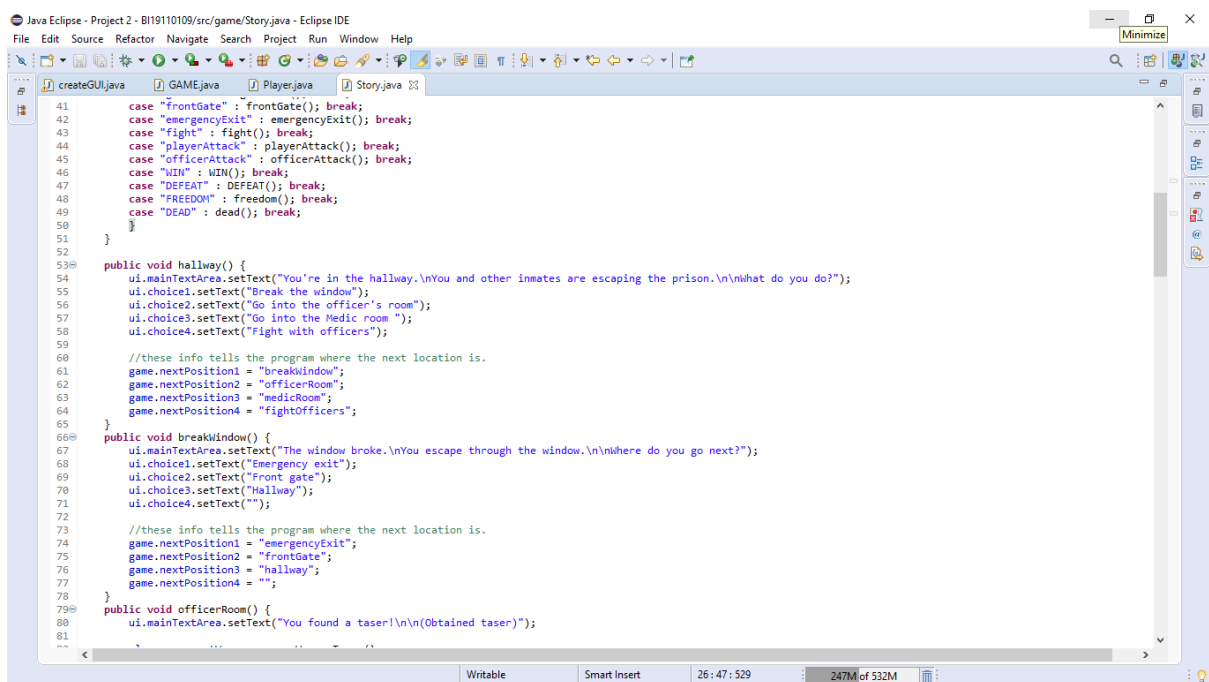
```
1 package game;
2
3 import game2.SuperWeapon;
4
5 public class Player {
6
7     public int HP;
8     public SuperWeapon currentWeapon;
9
10 }
11
```

## class Story



The screenshot shows the Eclipse IDE with the 'Story.java' file open. The code defines a package 'game', imports several classes from 'game2', and defines a public class 'Story' with a constructor and several methods including 'GAMESetup()' and 'selectPosition()'.

```
1 package game;
2
3 import game2.OfficerAttack;
4 import game2.SuperOfficer;
5 import game2.WeaponBaton;
6 import game2.WeaponTaser;
7
8 public class Story { //handle all the dialogues
9
10     GAME game;
11     createGUI ui;
12     VisibilityManager vm;
13     Player player = new Player();
14     SuperOfficer officer;
15
16     public Story(GAME g, createGUI userInterface, VisibilityManager vManager) {
17
18         game = g;
19         ui = userInterface;
20         vm = vManager;
21     }
22
23     public void GAMESetup() {
24
25         player.HP = 15;
26         ui.hLabelNum.setText("" + player.HP);
27
28         player.currentWeapon = new WeaponBaton();
29         ui.weaponLabelName.setText(player.currentWeapon.name);
30     }
31
32     public void selectPosition(String nextPosition) {
33
34         switch(nextPosition) {
35             case "hallway" : hallway(); break;
36             case "breakWindow" : breakWindow(); break;
37             case "officerRoom" : officerRoom(); break;
38             case "medicRoom" : medicRoom(); break;
39             case "fightOfficers" : fightOfficers(); break;
40             case "gameOver" : gameOver(); break;
41             case "frontGate" : frontGate(); break;
42             case "emergencyExit" : emergencyExit(); break;
43             case "WIN" : WIN(); break;
44             case "DEFEAT" : DEFEAT(); break;
45             case "FREEDOM" : freedom(); break;
46             case "DEAD" : dead(); break;
47         }
48     }
49
50     public void hallway() {
51
52         ui.mainTextArea.setText("You're in the hallway.\nYou and other inmates are escaping the prison.\nWhat do you do?");
53         ui.choice1.setText("Break the window");
54         ui.choice2.setText("Go into the officer's room");
55         ui.choice3.setText("Go into the Medic room");
56         ui.choice4.setText("Fight with officers");
57
58         //these info tells the program where the next location is.
59         game.nextPosition1 = "breakWindow";
60         game.nextPosition2 = "officerRoom";
61         game.nextPosition3 = "medicRoom";
62         game.nextPosition4 = "fightOfficers";
63     }
64
65     public void breakWindow() {
66
67         ui.mainTextArea.setText("The window broke.\nYou escape through the window.\nWhere do you go next?");
68         ui.choice1.setText("Emergency exit");
69         ui.choice2.setText("Front gate");
70         ui.choice3.setText("Hallway");
71         ui.choice4.setText("");
72
73         //these info tells the program where the next location is.
74         game.nextPosition1 = "emergencyExit";
75         game.nextPosition2 = "frontGate";
76         game.nextPosition3 = "hallway";
77         game.nextPosition4 = "";
78     }
79
80     public void officerRoom() {
81
82         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
83     }
84
85     public void medicRoom() {
86
87         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
88     }
89
90     public void fightOfficers() {
91
92         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
93     }
94
95     public void gameOver() {
96
97         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
98     }
99
100     public void frontGate() {
101
102         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
103     }
104
105     public void emergencyExit() {
106
107         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
108     }
109
110     public void WIN() {
111
112         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
113     }
114
115     public void DEFEAT() {
116
117         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
118     }
119
120     public void freedom() {
121
122         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
123     }
124
125     public void dead() {
126
127         ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
128     }
129
130 }
```



The screenshot shows the Eclipse IDE with the 'Story.java' file open. The code continues from the previous screenshot, defining methods for 'hallway()', 'breakWindow()', 'officerRoom()', 'medicRoom()', 'fightOfficers()', 'gameOver()', 'frontGate()', 'emergencyExit()', 'WIN()', 'DEFEAT()', 'freedom()', and 'dead()'.

```
41     case "frontGate" : frontGate(); break;
42     case "emergencyExit" : emergencyExit(); break;
43     case "fight" : fight(); break;
44     case "playerAttack" : playerAttack(); break;
45     case "officerAttack" : officerAttack(); break;
46     case "WIN" : WIN(); break;
47     case "DEFEAT" : DEFEAT(); break;
48     case "FREEDOM" : freedom(); break;
49     case "DEAD" : dead(); break;
50 }
51
52 public void hallway() {
53
54     ui.mainTextArea.setText("You're in the hallway.\nYou and other inmates are escaping the prison.\nWhat do you do?");
55     ui.choice1.setText("Break the window");
56     ui.choice2.setText("Go into the officer's room");
57     ui.choice3.setText("Go into the Medic room");
58     ui.choice4.setText("Fight with officers");
59
60     //these info tells the program where the next location is.
61     game.nextPosition1 = "breakWindow";
62     game.nextPosition2 = "officerRoom";
63     game.nextPosition3 = "medicRoom";
64     game.nextPosition4 = "fightOfficers";
65 }
66
67 public void breakWindow() {
68
69     ui.mainTextArea.setText("The window broke.\nYou escape through the window.\nWhere do you go next?");
70     ui.choice1.setText("Emergency exit");
71     ui.choice2.setText("Front gate");
72     ui.choice3.setText("Hallway");
73     ui.choice4.setText("");
74
75     //these info tells the program where the next location is.
76     game.nextPosition1 = "emergencyExit";
77     game.nextPosition2 = "frontGate";
78     game.nextPosition3 = "hallway";
79     game.nextPosition4 = "";
80 }
81
82 public void officerRoom() {
83
84     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
85 }
86
87 public void medicRoom() {
88
89     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
90 }
91
92 public void fightOfficers() {
93
94     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
95 }
96
97 public void gameOver() {
98
99     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
100 }
101
102 public void frontGate() {
103
104     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
105 }
106
107 public void emergencyExit() {
108
109     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
110 }
111
112 public void WIN() {
113
114     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
115 }
116
117 public void DEFEAT() {
118
119     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
120 }
121
122 public void freedom() {
123
124     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
125 }
126
127 public void dead() {
128
129     ui.mainTextArea.setText("You found a taser!\n(Obtained taser)");
130 }
131
132 }
```

```
Java Eclipse - Project 2 - B19110109/src/game/Story.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

createGUI.java GAME.java Player.java Story.java
82 player.currentWeapon = new WeaponTaser();
83 ui.weaponLabelName.setText(player.currentWeapon.name);
84
85 ui.choice1.setText("Go back to the hallway");
86 ui.choice2.setText("");
87 ui.choice3.setText("");
88 ui.choice4.setText("");
89
90 //these info tells the program where the next location is.
91 game.nextPosition1 = "hallway";
92 game.nextPosition2 = "";
93 game.nextPosition3 = "";
94 game.nextPosition4 = "";
95
96 }
97 public void medicRoom() {
98     ui.mainTextArea.setText("You found an emergency kit!\nYou quickly dress your wounds.\n\n(HP recovered by 3)");
99     player.HP = player.HP + 3;
100     ui.hLabelNum.setText("" + player.HP);
101     ui.choice1.setText("Go back to the hallway");
102     ui.choice2.setText("");
103     ui.choice3.setText("");
104     ui.choice4.setText("");
105
106 //these info tells the program where the next location is.
107 game.nextPosition1 = "hallway";
108 game.nextPosition2 = "";
109 game.nextPosition3 = "";
110 game.nextPosition4 = "";
111 }
112 public void fightOfficers() {
113     ui.mainTextArea.setText("Officer 1 : Do you think you can fight us!\nOfficer 2 : Stop being delusional.\nOfficer 3 : Catch him!\n\nYou've been caught!");
114     player.HP = player.HP - player.HP ;
115     ui.hLabelNum.setText("" + player.HP);
116     ui.choice1.setText("GAME OVER -");
117     ui.choice2.setText("");
118     ui.choice3.setText("");
119     ui.choice4.setText("");
120
121 //these info tells the program where the next location is.
122 game.nextPosition1 = "gameOver";
123 game.nextPosition2 = "";
124 game.nextPosition3 = "";
125
126 }
127
128 }
129
130 }
131
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }
149
150 }
151
152 }
153
154 }
155
156 }
157
158 }
159
160 }
161
162 }
163
164 }
165
166 }
167
168 }
169
170 }
171
172 }
173
174 }
175
176 }
177
178 }
179
180 }
181
182 }
183
184 }
185
186 }
187
188 }
189
190 }
191
192 }
193
194 }
195
196 }
197
198 }
199
200 }
201
202 }
203
204 }
205
206 }
207
208 }
209
210 }
211
212 }
213
214 }
215
216 }
217
218 }
219
220 }
221
222 }
223
224 }
225
226 }
227
228 }
229
230 }
231
232 }
233
234 }
235
236 }
237
238 }
239
240 }
241
242 }
243
244 }
245
246 }
247
248 }
249
250 }
251
252 }
253
254 }
255
256 }
257
258 }
259
260 }
261
262 }
263
264 }
265
266 }
267
268 }
269
270 }
271
272 }
273
274 }
275
276 }
277
278 }
279
280 }
281
282 }
283
284 }
285
286 }
287
288 }
289
290 }
291
292 }
293
294 }
295
296 }
297
298 }
299
300 }
301
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }
313
314 }
315
316 }
317
318 }
319
320 }
321
322 }
323
324 }
325
326 }
327
328 }
329
330 }
331
332 }
333
334 }
335
336 }
337
338 }
339
340 }
341
342 }
343
344 }
345
346 }
347
348 }
349
350 }
351
352 }
353
354 }
355
356 }
357
358 }
359
360 }
361
362 }
363
364 }
365
366 }
367
368 }
369
370 }
371
372 }
373
374 }
375
376 }
377
378 }
379
380 }
381
382 }
383
384 }
385
386 }
387
388 }
389
390 }
391
392 }
393
394 }
395
396 }
397
398 }
399
400 }
401
402 }
403
404 }
405
406 }
407
408 }
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
680 }
681
682 }
683
684 }
685
686 }
687
688 }
689
690 }
691
692 }
693
694 }
695
696 }
697
698 }
699
700 }
701
702 }
703
704 }
705
706 }
707
708 }
709
710 }
711
712 }
713
714 }
715
716 }
717
718 }
719
720 }
721
722 }
723
724 }
725
726 }
727
728 }
729
730 }
731
732 }
733
734 }
735
736 }
737
738 }
739
740 }
741
742 }
743
744 }
745
746 }
747
748 }
749
750 }
751
752 }
753
754 }
755
756 }
757
758 }
759
760 }
761
762 }
763
764 }
765
766 }
767
768 }
769
770 }
771
772 }
773
774 }
775
776 }
777
778 }
779
780 }
781
782 }
783
784 }
785
786 }
787
788 }
789
790 }
791
792 }
793
794 }
795
796 }
797
798 }
799
800 }
801
802 }
803
804 }
805
806 }
807
808 }
809
810 }
811
812 }
813
814 }
815
816 }
817
818 }
819
820 }
821
822 }
823
824 }
825
826 }
827
828 }
829
830 }
831
832 }
833
834 }
835
836 }
837
838 }
839
840 }
841
842 }
843
844 }
845
846 }
847
848 }
849
850 }
851
852 }
853
854 }
855
856 }
857
858 }
859
860 }
861
862 }
863
864 }
865
866 }
867
868 }
869
870 }
871
872 }
873
874 }
875
876 }
877
878 }
879
880 }
881
882 }
883
884 }
885
886 }
887
888 }
889
890 }
891
892 }
893
894 }
895
896 }
897
898 }
899
900 }
901
902 }
903
904 }
905
906 }
907
908 }
909
910 }
911
912 }
913
914 }
915
916 }
917
918 }
919
920 }
921
922 }
923
924 }
925
926 }
927
928 }
929
930 }
931
932 }
933
934 }
935
936 }
937
938 }
939
940 }
941
942 }
943
944 }
945
946 }
947
948 }
949
950 }
951
952 }
953
954 }
955
956 }
957
958 }
959
960 }
961
962 }
963
964 }
965
966 }
967
968 }
969
970 }
971
972 }
973
974 }
975
976 }
977
978 }
979
980 }
981
982 }
983
984 }
985
986 }
987
988 }
989
990 }
991
992 }
993
994 }
995
996 }
997
998 }
999
1000 }
```

```
Java Eclipse - Project 2 - B19110109/src/game/Story.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

createGUI.java GAME.java Player.java Story.java
123 game.nextPosition3 = "";
124 game.nextPosition4 = "";
125
126 }
127 public void gameOver() {
128     ui.mainTextArea.setText("PLEASE CLOSE THE WINDOW AND TRY AGAIN .\n\nGOODLUCK! ^-^ ");
129 }
130 public void frontGate() {
131     ui.mainTextArea.setText("Officer 1 : Well, what do we have here.\nOfficer 2 : It's our lucky day huh.\nOfficer 3 : Catch him!\n\nThey shot your arm.You've been caught");
132     player.HP = player.HP - player.HP ;
133     ui.hLabelNum.setText("" + player.HP);
134     ui.choice1.setText("GAME OVER -");
135     ui.choice2.setText("");
136     ui.choice3.setText("");
137     ui.choice4.setText("");
138
139 //these info tells the program where the next location is.
140 game.nextPosition1 = "gameOver";
141 game.nextPosition2 = "";
142 game.nextPosition3 = "";
143 game.nextPosition4 = "";
144 }
145 public void emergencyExit() {
146     officer = new OfficerAttack(); //officer is a super class, so we put the "OfficerAttack" class in the super class
147
148     ui.mainTextArea.setText("You run into an " + officer.name + "!");
149     ui.choice1.setText("Run");
150     ui.choice2.setText("Fight");
151     ui.choice3.setText("");
152     ui.choice4.setText("");
153
154 //these info tells the program where the next location is.
155 game.nextPosition1 = "breakWindow";
156 game.nextPosition2 = "fight";
157 game.nextPosition3 = "";
158 game.nextPosition4 = "";
159 }
160 public void fight() {
161     ui.mainTextArea.setText(officer.name + " HP : " + officer.HP + "\n\nWhat do you do ?");
162     ui.choice1.setText("Run");
163     ui.choice2.setText("Attack");
164
165 }
166
167 }
168
169 }
170
171 }
172
173 }
174
175 }
176
177 }
178
179 }
180
181 }
182
183 }
184
185 }
186
187 }
188
189 }
190
191 }
192
193 }
194
195 }
196
197 }
198
199 }
200
201 }
202
203 }
204
205 }
206
207 }
208
209 }
210
211 }
212
213 }
214
215 }
216
217 }
218
219 }
220
221 }
222
223 }
224
225 }
226
227 }
228
229 }
230
231 }
232
233 }
234
235 }
236
237 }
238
239 }
240
241 }
242
243 }
244
245 }
246
247 }
248
249 }
250
251 }
252
253 }
254
255 }
256
257 }
258
259 }
260
261 }
262
263 }
264
265 }
266
267 }
268
269 }
270
271 }
272
273 }
274
275 }
276
277 }
278
279 }
280
281 }
282
283 }
284
285 }
286
287 }
288
289 }
290
291 }
292
293 }
294
295 }
296
297 }
298
299 }
300
301 }
302
303 }
304
305 }
306
307 }
308
309 }
310
311 }
312
313 }
314
315 }
316
317 }
318
319 }
320
321 }
322
323 }
324
325 }
326
327 }
328
329 }
330
331 }
332
333 }
334
335 }
336
337 }
338
339 }
340
341 }
342
343 }
344
345 }
346
347 }
348
349 }
350
351 }
352
353 }
354
355 }
356
357 }
358
359 }
360
361 }
362
363 }
364
365 }
366
367 }
368
369 }
370
371 }
372
373 }
374
375 }
376
377 }
378
379 }
380
381 }
382
383 }
384
385 }
386
387 }
388
389 }
390
391 }
392
393 }
394
395 }
396
397 }
398
399 }
400
401 }
402
403 }
404
405 }
406
407 }
408
409 }
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
680 }
681
682 }
683
684 }
685
686 }
687
688 }
689
690 }
691
692 }
693
694 }
695
696 }
697
698 }
699
700 }
701
702 }
703
704 }
705
706 }
707
708 }
709
710 }
711
712 }
713
714 }
715
716 }
717
718 }
719
720 }
721
722 }
723
724 }
725
726 }
727
728 }
729
730 }
731
732 }
733
734 }
735
736 }
737
738 }
739
740 }
741
742 }
743
744 }
745
746 }
747
748 }
749
750 }
751
752 }
753
754 }
755
756 }
757
758 }
759
760 }
761
762 }
763
764 }
765
766 }
767
768 }
769
770 }
771
772 }
773
774 }
775
776 }
777
778 }
779
780 }
781
782 }
783
784 }
785
786 }
787
788 }
789
790 }
791
792 }
793
794 }
795
796 }
797
798 }
799
800 }
801
802 }
803
804 }
805
806 }
807
808 }
809
810 }
811
812 }
813
814 }
815
816 }
817
818 }
819
820 }
821
822 }
823
824 }
825
826 }
827
828 }
829
830 }
831
832 }
833
834 }
835
836 }
837
838 }
839
840 }
841
842 }
843
844 }
845
846 }
847
848 }
849
850 }
851
852 }
853
854 }
855
856 }
857
858 }
859
860 }
861
862 }
863
864 }
865
866 }
867
868 }
869
870 }
871
872 }
873
874 }
875
876 }
877
878 }
879
880 }
881
882 }
883
884 }
885
886 }
887
888 }
889
890 }
891
892 }
893
894 }
895
896 }
897
898 }
899
900 }
901
902 }
903
904 }
905
906 }
907
908 }
909
910 }
911
912 }
913
914 }
915
916 }
917
918 }
919
920 }
921
922 }
923
924 }
925
926 }
927
928 }
929
930 }
931
932 }
933
934 }
935
936 }
937
938 }
939
940 }
941
942 }
943
944 }
945
946 }
947
948 }
949
950 }
951
952 }
953
954 }
955
956 }
957
958 }
959
960 }
961
962 }
963
964 }
965
966 }
967
968 }
969
970 }
971
972 }
973
974 }
975
976 }
977
978 }
979
980 }
981
982 }
983
984 }
985
986 }
987
988 }
989
990 }
991
992 }
993
994 }
995
996 }
997
998 }
999
1000 }
```

```
Java Eclipse - Project 2 - BI19110109/src/game/Story.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

createGUI.java GAME.java Player.java Story.java

164 ui.choice2.setText("Attack");
165 ui.choice3.setText("");
166 ui.choice4.setText("");
167
168 //these info tells the program where the next location is.
169 game.nextPosition1 = "hallway";
170 game.nextPosition2 = "playerAttack";
171 game.nextPosition3 = "";
172 game.nextPosition4 = "";
173 }
174 public void playerAttack() {
175     int playerDamage = 0 ;
176
177     //to differentiate the damage based on what weapon you own
178     if(player.currentWeapon.name.equals("Baton")) {
179         playerDamage = player.currentWeapon.damage;
180     }
181     else if (player.currentWeapon.name.equals("Taser")) {
182         playerDamage = player.currentWeapon.damage;
183     }
184 }
185
186 ui.mainTextArea.setText("You attack the officer and gave " + playerDamage + " damage!");
187
188 officer.HP = officer.HP - playerDamage;
189
190 ui.choice1.setText(">");
191 ui.choice2.setText("");
192 ui.choice3.setText("");
193 ui.choice4.setText("");
194
195 //need to make if-else statement to know if the officer is still alive or not
196 if(officer.HP>0) {
197     game.nextPosition1 = "officerAttack";
198     game.nextPosition2 = "";
199     game.nextPosition3 = "";
200     game.nextPosition4 = "";
201 }
202 else if(officer.HP<1) {
203     game.nextPosition1 = "WIN";
204     game.nextPosition2 = "";
205 }
```

```
Java Eclipse - Project 2 - BI19110109/src/game/Story.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

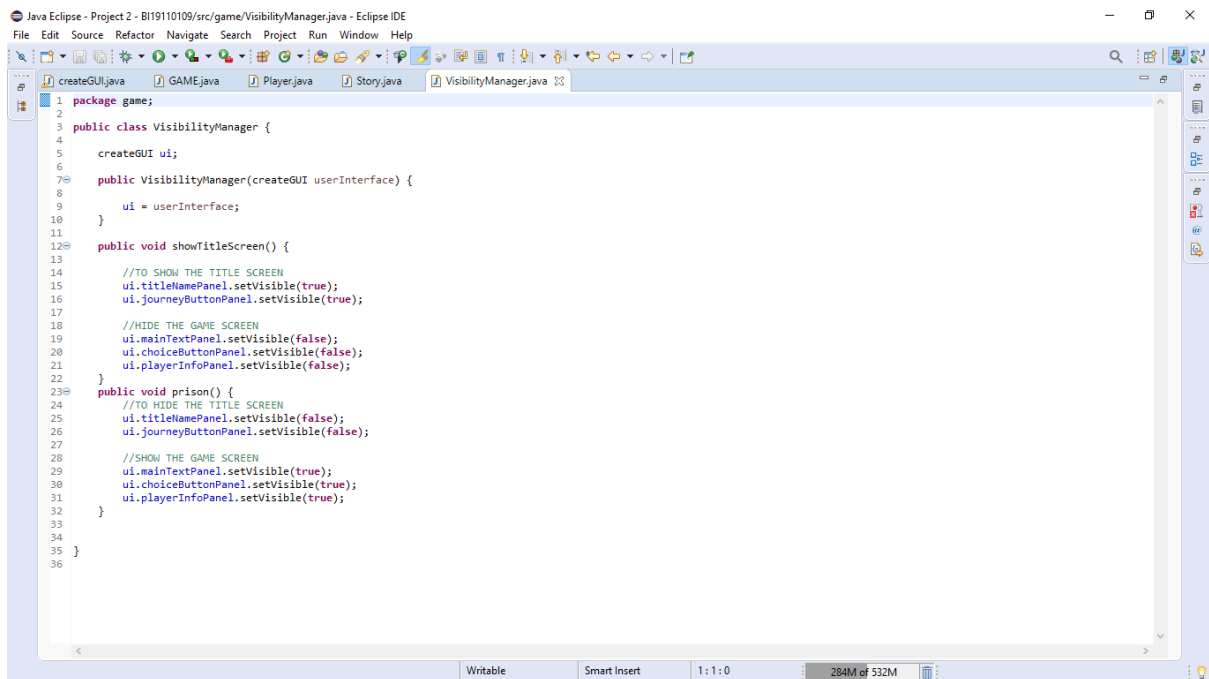
createGUI.java GAME.java Player.java Story.java

204 game.nextPosition2 = "";
205 game.nextPosition3 = "";
206 game.nextPosition4 = "";
207 }
208 }
209 public void officerAttack() {
210     ui.mainTextArea.setText("The officer attack you back.\n\n(Recieved " + officer.attack + " damage)");
211     player.HP = player.HP - officer.attack;
212     ui.hLabelNum.setText("" + player.HP);
213
214     ui.choice1.setText(">");
215     ui.choice2.setText("");
216     ui.choice3.setText("");
217     ui.choice4.setText("");
218
219 //need to make if-else statement to know if the player is still alive or not
220 if(player.HP>0) {
221     game.nextPosition1 = "fight";
222     game.nextPosition2 = "";
223     game.nextPosition3 = "";
224     game.nextPosition4 = "";
225 }
226 else if(player.HP<1) {
227     game.nextPosition1 = "DEFEAT";
228     game.nextPosition2 = "";
229     game.nextPosition3 = "";
230     game.nextPosition4 = "";
231 }
232 }
233 }
234 }
235 public void WIN() {
236     ui.mainTextArea.setText("You defeated the officer!\n\nThe officer dropped the key for the Emergency exit.\n\n(Obtained Key)");
237     game.nextPosition1 = "FREEDOM";
238     ui.choice1.setText("freedom");
239 }
240 }
241 public void DEFEAT() {
242     ui.mainTextArea.setText("You're dead!\n");
243 }
```

```
244 ui.mainTextArea.setText("You're dead!\n");
245 game.nextPosition1 = "DEAD";
246 ui.choice1.setText("dead");
247 }
248 }
249 }
250 public void freedom() {
251     ui.mainTextArea.setText("\nTHE DOOR UNLOCKED!\nCONGRATULATIONS! YOU'VE MADE YOUR WAY TO FREEDOM (^-^)/.\n\nPLEASE CLOSE THE WINDOW TO EXIT");
252 }
253 }
254 public void dead() {
255     ui.mainTextArea.setText("YOU'VE BEEN KILLED!\nPLEASE CLOSE THE WINDOW AND TRY AGAIN .\n\nGOODLUCK! ^-^ ");
256 }
257 }
258 }
```



## Class VisibilityManager




The screenshot shows the Eclipse IDE with the file `VisibilityManager.java` open. The code is as follows:

```
1 package game;  
2  
3 public class VisibilityManager {  
4     createGUI ui;  
5  
6     public VisibilityManager(createGUI userInterface) {  
7  
8         ui = userInterface;  
9     }  
10  
11  
12     public void showTitleScreen() {  
13  
14         //TO SHOW THE TITLE SCREEN  
15         ui.titleNamePanel.setVisible(true);  
16         ui.journeyButtonPanel.setVisible(true);  
17  
18         //HIDE THE GAME SCREEN  
19         ui.mainTextPanel.setVisible(false);  
20         ui.choiceButtonPanel.setVisible(false);  
21         ui.playerInfoPanel.setVisible(false);  
22     }  
23  
24     public void prison() {  
25         //TO HIDE THE TITLE SCREEN  
26         ui.titleNamePanel.setVisible(false);  
27         ui.journeyButtonPanel.setVisible(false);  
28  
29         //SHOW THE GAME SCREEN  
30         ui.mainTextPanel.setVisible(true);  
31         ui.choiceButtonPanel.setVisible(true);  
32         ui.playerInfoPanel.setVisible(true);  
33     }  
34  
35 }  
36
```

Package = game2

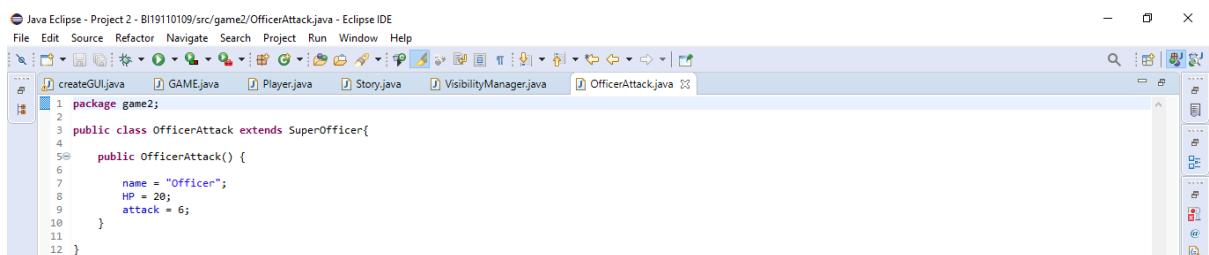
class SuperOfficer



The screenshot shows the Eclipse IDE with the file `SuperOfficer.java` open. The code is as follows:

```
1 package game2;  
2  
3 public class SuperOfficer {  
4  
5     //constructors  
6     public String name;  
7     public int HP;  
8     public int attack;  
9  
10  
11 }  
12
```

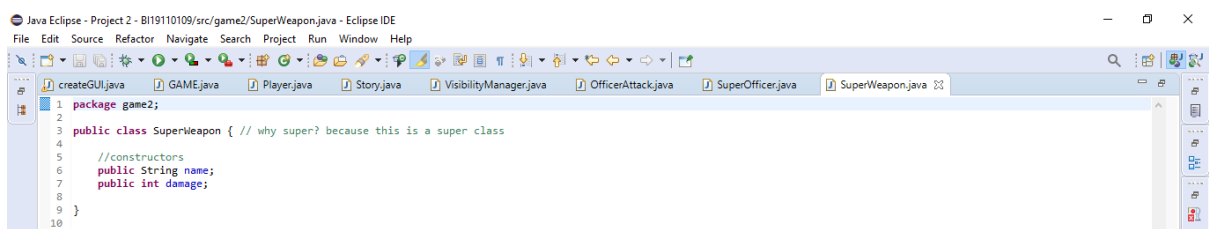
class OfficerAttack



The screenshot shows the Eclipse IDE with the file `OfficerAttack.java` open. The code is as follows:

```
1 package game2;  
2  
3 public class OfficerAttack extends SuperOfficer{  
4  
5     public OfficerAttack() {  
6  
7         name = "Officer";  
8         HP = 20;  
9         attack = 6;  
10     }  
11  
12 }  
13
```

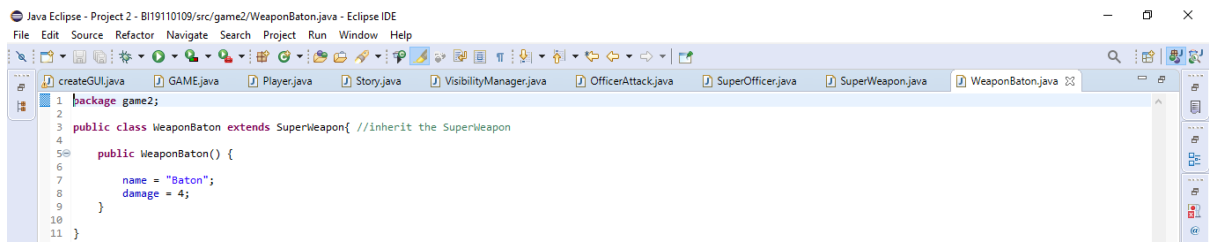
class SuperWeapon



The screenshot shows the Eclipse IDE with the file `SuperWeapon.java` open. The code is as follows:

```
1 package game2;  
2  
3 public class SuperWeapon { // why super? because this is a super class  
4  
5     //constructors  
6     public String name;  
7     public int damage;  
8  
9 }  
10
```

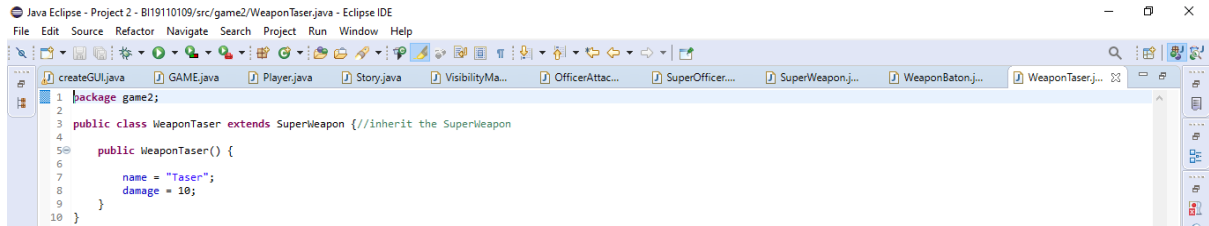
## class WeaponBaton



The screenshot shows the Eclipse IDE interface with the 'WeaponBaton.java' file open. The title bar indicates the project is 'Project 2 - B119110109/src/game2/WeaponBaton.java - Eclipse IDE'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development tools. The package explorer on the left shows a project structure with files like createGUI.java, GAME.java, Player.java, Story.java, VisibilityManager.java, OfficerAttack.java, SuperOfficer.java, SuperWeapon.java, and WeaponBaton.java. The editor window displays the following Java code:

```
1 package game2;
2
3 public class WeaponBaton extends SuperWeapon { //inherit the SuperWeapon
4
5     public WeaponBaton() {
6
7         name = "Baton";
8         damage = 4;
9     }
10
11 }
```

## Class WeaponTaser



The screenshot shows the Eclipse IDE interface with the 'WeaponTaser.java' file open. The title bar indicates the project is 'Project 2 - B119110109/src/game2/WeaponTaser.java - Eclipse IDE'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development tools. The package explorer on the left shows a project structure with files like createGUI.java, GAME.java, Player.java, Story.java, VisibilityMa..., OfficerAttac..., SuperOfficer..., SuperWeapon.j..., WeaponBaton.j..., and WeaponTaser.j.... The editor window displays the following Java code:

```
1 package game2;
2
3 public class WeaponTaser extends SuperWeapon { //inherit the SuperWeapon
4
5     public WeaponTaser() {
6
7         name = "Taser";
8         damage = 10;
9     }
10 }
```

## OBJECT ORIENTED PROGRAMMING CONCEPT

### 1. Inheritance

- Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object.
- It is the ability to create a new class from an existing class.
  - Parent class (Superclass) is the general class i.e. SuperWeapon, SuperOfficer.
  - Children classes (subclass) is the more specific classes i.e. WeaponBaton, WeaponTaser, OfficerAttack.
  - These common classes share many attributes and behaviors (code reusability)

### 2. Abstraction

- Abstraction is a process of hiding the implementation details and showing only functionality to the user.
- It is the ability to represent data at a very conceptual level without any details.
- Another way, it shows only essential things to the user and hides the internal details, for example, clicking the buttons in the game. User does not know the internal processing about the next action will be.
- Abstraction lets you focus on what the object does instead of how it does it.

### 3. Polymorphism

- Polymorphism in Java is a concept by which we can perform a single action in different ways.
- A single statement is used to call various methods and the exact method to be called is only known during runtime.