```java
1
2 /* This class encapsulates Monster position and direction
3  * It also keeps a reference to the player it is tracking
4  * The canView attribute can be used to limit monster visibility
5  */
6
7 public class Monster extends Moveable {
8     private boolean canView = true; // allows
9     private Player player;
10    boolean isBaby = false;
11    int bornTime = 0;
12
13    public Monster(Grid g, Player p, int row, int col) throws
  Exception {
14        super(g);
15        player = p;
16        setCell(grid.getCell(row, col));
17    }
18
19    // check if it's baby monster
20    public boolean isBaby() {
21        return isBaby;
22    }
23
24    public Cell move(int presses) {
25        boolean gotStoped = false;
26        if (bornTime > 0) {
27            for (int i = 0; i < player.getBlock().size(); ++i) {
28                if (player.getBlock().get(i).getHealth() == 0) {
29                    player.getBlock().remove(i);
30                } else if (currentCell ==
  player.getBlock().get(i).getCell()) {
31                    player.getBlock().get(i).healthDec();
32                    gotStoped = true;
33                }
34            }
35            for (int i = 0; i < player.getTrap().size(); ++i) {
36                if (currentCell ==
  player.getTrap().get(i).getCell()) {
37                    gotStoped = true;
38                }
39            }
```

```java
40              if (gotStoped == false) {
41                  if (currentCell.row % 5 == 0 && currentCell.col %
    5 == 0
42                      && (currentCell.row ==
    player.getCell().row || currentCell.col == player.getCell().col))
    {
43                      currentCell = player.getCell();
44                  } else {
45                      currentDirection =
    grid.getBestDirection(currentCell, player.getCell());
46                      currentCell = grid.getCell(getCell(),
    getDirection(), 1);
47                  }
48              }
49          }
50      bornTime++;
51      if (bornTime >= 10)
52          isBaby = false;
53      return currentCell;
54  }
55
56  // check if monster needs to be hidden
57  public boolean viewable() {
58      Cell p = player.getCell();
59      Cell m = currentCell;
60      if (bornTime > 0 && (Math.abs(p.col - m.col) > 4 ||
    Math.abs(p.row - m.row) > 4)) {
61          canView = false;
62      } else {
63          canView = true;
64      }
65      return canView;
66  }
67 }
68
```