Breakout openHPI Java-Konzept

Klassendiagramme (4 Klassen)

Spielfeld
int breite
int hoehe
JPanel felder[breite][hoehe] //JPanel sind kleine Bedienelemente, auf denen man etwas darstellen kann
double zeit
int punktzahl
int anzahlBaelle
int anzahlMauern
void generiereSpielfeld()
void spielende()
void ballWeniger()
void mauerWeniger()
void addPunkte(int punkte)
int getBreite()
int getHoehe()

Paddle
int x
int y
int breite
void verschiebe()
int getX()
int getY()
int getLaenge()

Mauer
int x
int y
int breite
int punktzahl
boolean zerstoert
boolean isZerstoert()
void setZerstoert()
int getPunktzahl()

Ball
int x
int y
int flugrichtung //oben Rechts =1;unten Rechts =2; unten Links=3;oben Links =4
void pralleAb()
void fliegeweiter()

Quellcodeentwurf für alle 4 Klassen

Spielfeld

```
1  package breakout;
2
3  import javax.swing.JPanel;
4
5  public class Spielfeld {
6
7      private int breite = 30; //Breite und Höhe des Spielfeldes in Zelleinheiten wird festgelegt
8      private int hoehe = 20;
9      private JPanel[][] felder = new JPanel[breite][hoehe]; //Die Zellen werden erstellt
10     private double zeit = 0; //Die Spielzeit und die Punktzahl werden erstelt
11     private int punktzahl = 0;
12     private int anzahlBaelle = 3; //Jeder Spieler hat 3 Bälle zum Spielen
13     private int anzahlMauern = 15; //Es existieren 3 Reihen zu je 5 Mauern
14
15     public Spielfeld() {
16         new Paddle(12, 17, 6);
17         generiereSpielfeld();
18     }
19
20     public void generiereSpielfeld() { //Die 3 mal 5 Mauern werden hier erstellt
21         for(int y=0;y<5;y++) {
22             for(int x=0;x<3;x++) {
23                 felder[x][y] = new Mauer(1+5*(x-1),2+y,5);
24             }
25         }
26     }
27
28     public void spielende() {
29         //Wenn der Spieler keine Bälle mehr hat, verliert er
30         if(anzahlBaelle==0) {
31             System.out.println("Deine Bälle sind aufgebraucht. Du hast verloren.");
32             System.out.println("Deine Spielzeit betrug: "+zeit);
33             System.out.println("Du hast "+punktzahl+" Punkte erreicht.");
34         //Wenn der Spieler alle Mauern entfernt hat, gewinnt er
35         } else if(anzahlMauern==0) {
36             System.out.println("Du hast alle Mauern entfernt. Du hast gewonnen.");
37             System.out.println("Deine Spielzeit betrug: "+zeit);
38             System.out.println("Du hast "+punktzahl+" Punkte erreicht.");
39         }
40     }
41
42     public void ballWeniger() { //Wenn ein Ball ins Aus geht, hat der Spieler einen Ball weniger
43         anzahlBaelle--;
44     }
45
46     public void mauerWeniger() { //Wenn eine Mauer zerstört ist, geht der Zähler eins runter
47         anzahlMauern--;
48     }
49
50     public void addPunkte(int punkte) { //Hier werden Punkte be Mauerzerstörung addiert
51         punktzahl += punkte;
52     }
53
54     public int getBreite() { //Get für die Hoehe und Breite
55         return breite;
56     }
57
58     public int getHoehe() {
59         return hoehe;
60     }
61
62     public static void main(String[] args) { //Der Start des Programmes
63         new Spielfeld();
64     }
65
66 }
```

Paddle

```java
package breakout;

public class Paddle {

    private int x; //Das Paddle hat eine Startposition von x, y
    private int y;
    private int laenge; //Die Länge geht ab (x,y) laenge Positionen nach Rechts

    public Paddle(int x, int y, int laenge) {
        this.x = x;
        this.y = y;
        this.laenge = laenge;
    }

    public void verschiebe() {
        //Wenn der Spieler nach Links klickt, geht er nach links und andersherum
        //Linksklick und Rechtsklick sind hier provisorische Befehle. Man benötigt KeyListener
        if(linksklick && x > 0) {
            x--;
        } else if(rechtsklick && x < new Spielfeld().getBreite()) {
            x++;
        }
    }

    public int getX() { //Die Position des Paddles wird gegeben
        return x;
    }

    public int getY() {
        return y;
    }

    public int getLaenge() {
        return laenge;
    }



}
```

Mauer

```java
package breakout;

import javax.swing.JPanel;

public class Mauer extends JPanel {

    int x,y,breite; //Die Mauer ist wie das Paddle von (x,y) bis (x+breite,y) groß
    int punktzahl = 10;
    boolean zerstoert = false; //Ein Boolean, ob die Mauer noch existiert oder nicht

    public Mauer(int x, int y, int breite) {
        this.x = x;
        this.y = y;
        this.breite = breite;
    }

    public boolean isZerstoert() { //Man erfährt, ob die Mauer noch da ist
        return zerstoert;
    }

    public void setZerstoert(boolean zerstoert) { //Stellt um, ob die Mauer existiert oder nicht
        this.zerstoert = zerstoert;
    }

    public int getPunktzahl() { //Hier Addiert man die Punkte, wenn die Mauer zerstört ist
        return punktzahl;
    }

}

```

Ball

```java
package breakout;

public class Ball {

    private int x, y; //Der Ball hat einen x und y Wert als Position
    private int flugrichtung; //nach: oben Rechts =1;unten Rechts =2; unten Links=3;oben Links =4

    public Ball(int x, int y, int richtung) {
        this.x = x;
        this.y = y;
        this.flugrichtung = richtung;
    }

    public void pralleAb() {
        Spielfeld spf = new Spielfeld();
        int breite = spf.getBreite();
        int hoehe = spf.getHoehe();
        Paddle pd = new Paddle(12, 17, 6);

        //Je nach Flugrichtung und wo er gegenprallt, wird die Flugrichtung angepasst
        if(x==0 && flugrichtung==3) {
            flugrichtung = 2;
        } else if(x==0 && flugrichtung==4) {
            flugrichtung = 1;
        } else if(x==breite && flugrichtung==1) {
            flugrichtung = 4;
        } else if(x==breite && flugrichtung==2) {
            flugrichtung = 3;
        } else if(y==0 && flugrichtung==1) {
            flugrichtung = 2;
        } else if(y==0 && flugrichtung==4) {
            flugrichtung = 3;
        } else if(y==pd.getY() && flugrichtung==2) {
            flugrichtung = 1;
        } else if(y==pd.getY() && flugrichtung==3) {
            flugrichtung = 4;
        } else if(y>hoehe) {
            System.out.println("Der Ball ist ins Aus gegangen.");
            spf.ballWeniger();
        }
    }

    public void fliegeWeiter() {
        //Je nach Flugrichtung, fliegt er auch nach (x,y)
        if(flugrichtung == 1) {
            x++;
            y--; //in Java beginnt ein Koordinatensystem oben Links
        } else if (flugrichtung == 2) {
            x++;
            y++;
        } else if (flugrichtung == 3) {
            x--;
            y++;
        } else if (flugrichtung == 4) {
            x--;
            y--;
        }

        //Wen die Koordinaten des Balls denen einer Mauer entspricht, dann wird Mauer.setZerstoert(true) aufgerufen
        //außerdem gilt dann spf.mauerWeniger()
        //spf.addPunkte(Mauer.getPunkte); Dies ergänzt dann auch dem Punktestand die Punkte selbiger Mauer
    }

}
```