

Uppgift 3 - Lek med geometri

Författare:

Alexander Lindgren

alexander2000alle@gmail.com

Datum: 02/04-2024

JDK 17

Figurer

Klasserna:

- Triangele
- Square
- Circle

Triangle är en klass som representerar en triangel. Den har instansvariabler för triangelns position (x, y) och höjd (h), samt färgen (color). Metoden draw(Graphics g) ritar triangeln med angiven färg. Metoden contains(int x, int y) kontrollerar om en given punkt är inuti triangeln. Metoden move(int dx, int dy) flyttar triangeln med ett visst förskjutningsvärde. Metoden getBoundingBox() returnerar en rektangel som omsluter triangeln. Metoden getColor() returnerar färgen på triangeln. Den har också en (Polygon) som i Java är en inbyggd klass som representerar en polygon (en mångsidig geometrisk form med raka sidor) som kommer hjälpa oss med triangelns sidor.

Square-klassen tillåter användaren att skapa och hantera kvadratformade figurer. Den ger en mer specifik representation av en kvadrat med attribut som position (x, y), sidolängd och färg. Genom att implementera metoder som draw, contains, move, getBoundingBox och getColor, möjliggör den användning av kvadratformade figurer på en panel, och ger också möjligheten att interagera med och manipulera dem.

Circle-klassen tillåter användaren att skapa och hantera cirkelformade figurer. Den ger en bättre representation av en cirkel med attribut som position (x, y), radie (radius) och färg (color).

Alla dessa klasser (Triangle, Square, och Circle) implementerar gränssnittet Shape, vilket innebär att de måste tillhandahålla metoder som draw, contains, move, getBoundingBox, och getColor. Detta gör det möjligt att behandla dessa olika typer av geometriska figurer enhetligt och underlättar hanteringen av dem i ett gemensamt sammanhang, t.ex. i en gemensam lista av figurer.

Få figurerna att hamna överst

Inom Shape-gränssnittet definierar vi metoder för att rita, kontrollera innehåll och flytta figuren. Eftersom varje figur implementerar detta gränssnitt kan vi hantera alla figurer på samma sätt. I DrawingPanel-klassen håller vi en array av figurer (shapes) och en referens till den valda figuren (selectedShape). När musen trycks ned, hittar vi den valda figuren och sparar dess index. När musen släpps, ritar vi om panelen för att visa den uppdaterade positionen.

Metoden getShapeAtPoint i DrawingPanel-klassen används för att hitta den översta figuren på den plats där musen klickades. Den loopar igenom figurerna baklänges och returnerar den översta figuren som innehåller den givna punkten (x, y). Metoden paintComponent i DrawingPanel-klassen är ansvarig för att rita alla figurer. Först loopar den igenom alla figurer för att rita dem, förutom den valda figuren (selectedShape). Därefter ritar den den valda figuren, om det finns någon, vilket gör att den hamnar överst. I mousePressed-metoden i DrawingPanel-klassen hittar vi den valda figuren som musen klickade på och sparar dess position relativt musens position. Detta gör att figuren flyttas korrekt när musen dras. Sedan i mouseDragged-metoden i DrawingPanel-klassen flyttar vi den valda figuren relativt musens position när den dras. Eftersom vi alltid ritar den valda figuren sist, kommer den att hamna överst på de andra figurerna när den flyttas.

Utöka med fler figurer

Skapa nya instanser av Square och Triangle eller Circle med önskade parametrar (position, storlek, färg) i konstruktorn. Lägg till de nya figurerna i arrayen shapes. Testa gärna med dessa eller gör dina egna:

```
new Triangle(150, 50, 80, Color.PINK), // En rosa triange
new Square(300, 50, 80, Color.GREEN), // En grön kvadrat
new Circle(450, 90, 40, Color.ORANGE) // En orange cirkel
```