|  |
| --- |
| Högskolan i Gävle |
| Miniprojekt 4 |
| State och Decorator |
|  |
| *Hanna Medén, Niklas Nordgren* |
|  |
| 2020-02-16 |

|  |
| --- |
| Miniprojekt 4 i kursen Objektorienterad design och programmering II, DVG503 |
|  |

**Innehållsförteckning**

[1 Inledning 1](#_Toc32171435)

[2 Metod 2](#_Toc32171436)

[3 Resultat 5](#_Toc32171437)

[4 Diskussion 6](#_Toc32171438)

[5 Bilaga – UML för systemet 7](#_Toc32171439)

1. Inledning

I det här projektet modifierades gammal kod som övertogs av kursansvarige för att implementera designmönstret State. Koden som ärvdes innehöll designmönstret Decorator, som i detta fall implementerades med hjälp av en 2D cirkel som under körning kan ändra utseende genom att använda en grafisk dekoration som fyller cirkeln med svart färg. Målet var att utöka koden så att det fanns fler geometriska figurer att rita ut, samt fler dekorationer för att ändra utseendet av figurerna.

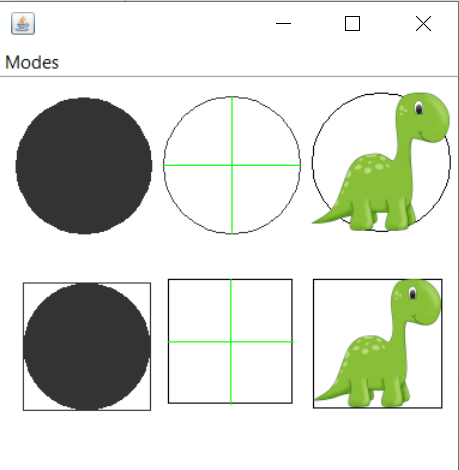
1. Metod

Inledningsvis implementerades utökningar av det befintliga designmönstret Decorator. Från början fanns klassen ShapeDecorator som implementerar interface:et Shape, detta interface implementeras även av de geometriska 2D figurerna, som ursprungligen endast var klassen Circle. Detta betyder att klassen ShapeDecorator och Circle har den gemensamma typen Shape. Vidare tar ShapeDecorator in ett objekt av typen Shape som argument i konstruktorn, detta för att utföra sina egna extra operationer, som i just detta fall är att fylla den grafiska representationen av ett Shape objekt med en svart oval, för att sedan delegera vidare till det insvepta objektet som togs emot vid initiering i konstruktorn.

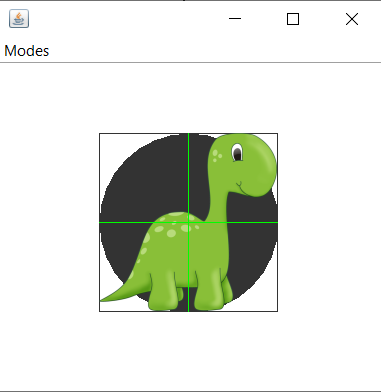
Klassen Rectangle adderades, likt klassen Circle implementerar den interface:et Shape för att möjliggöra dekorationer under körning. Inte helt oväntat så representerar objekt typen Rectangle grafiska 2D rektanglar, som vidare kan dekoreras tack vare Decorator designmönstret. Två klasser som används för att dekorera 2D figurer skapades, ShapeDecoratiorCrosshair och ShapeDecoratorDino, där den förstnämnda dekorerar en grafisk 2D figur med ett hårkors och den sistnämnda med en bild av en dinosaurie.

1. Resultat

Nedan i Figur 1 visas programmet under exekvering där sex geometriska 2D figurer ritats ut och dekorerats med ett Decorator-objekt vardera. Vidare i Figur 2 visas den grafiska representationen av ett objekt av typen Rectangle som dekorerats med tre olika dekorations objekt.



Figur 1: Programmet vid exekvering. 2D figurer en decoration vardera.



Figur 2: Programmet vid exekvering. Rektangel med staplade dekorationer.

1. Diskussion

Efter genomfört projekt har vi fått förståelse för att det kan finnas stora fördelar med att använda designmönstrena Decorator och State då de ger applikationer större flexibilitet samt underlättar underhållsarbetet vid utökning/ändring av koden.

1. Bilaga – UML för systemet

