|  |
| --- |
| 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 för att sedan delegera vidare till det insvepta objektet som togs emot vid initiering i konstruktorn.

Klassen Rectangle adderades till samlingen av geometriska 2D figurer, 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.

1. Resultat
2. Diskussion
3. Bilaga – UML för systemet