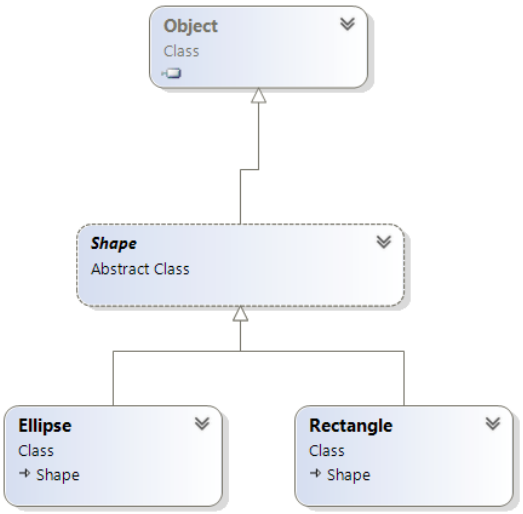
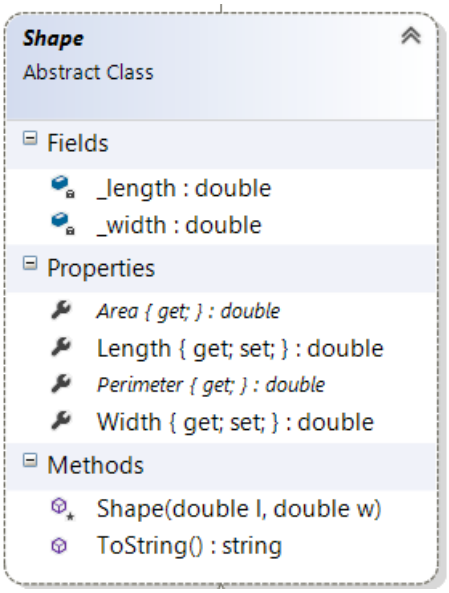
Objektorientering

Abstracta klasser och Arv

I den här uppgiften ska du skriva ett enkelt program som kan hantera Rektanglar och Ellipser. De båda geometriska figurerna har vissa gemensamma attribut och metoder och dessa kan därför implementeras som en abstrakt basklass som vi kallar Shape från vilken de båda geometriska figurerna ärver. **Observera att vissa namn på metoder och typer kan vara fel i klassdiagrammen då dessa kommer från C# och inte Java! Tex står Main och ToString med stor bokstav men i Java använder vi liten bokstav.**



Klassen Shape

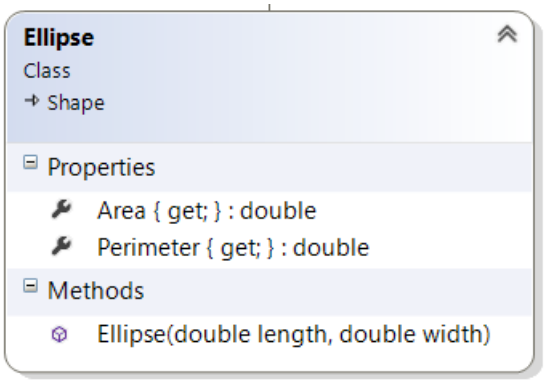


Shape är abstrakt och det kommer därför inte gå att skapa objekt av denna typ. Endast subklasserna Ellipse och Rectangle kommer gå att skapa objekt ifrån. Dessa båda klasser måste också implementera Area och Perimeter som är abstrakta properties.

Metoden toString
Publik metod som overrides metoden toString() i basklassen Object.
Strängen som returneras ska vara på formatet:

Längd : Length
Bredd : Width
Omkrets: Perimeter
Area : Area

Klassen Ellipse

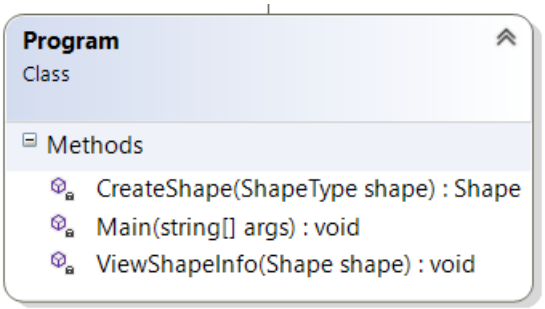


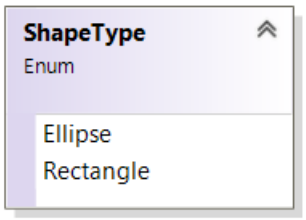
Konstruktorn
Låt klassens konstruktor anropa basklassens konstruktor för att sätta längd och bredd. Implementera även Area och Perimeter.

Klassen Rectangle

Implementeras på motsvarande sätt som Ellipse.

Program klass
I klassen program hittar vi main-metoden samt ett antal metoder för att hantera utskrifter och skapande av objekt i programmet. Utöver metoderna som visas i diagrammet nedan kan du även behöva lägga dit egna metoder för utritning och inläsning.





Metoden CreateShape

Används för att skapa nya objekt av önskad typ. Typen anges genom att skicka in ett enum-värde till metoden. Boken kap 12. sid 410

Metoden Main
Startpunkten för programmet som ska visa en meny med valmöjligheten att skapa någon av de typer som programmet stödjer samt att avsluta programmet.
När ett val har gjorts ska ViewShapeInfo anropas för det skapade objektet.

Metoden ViewShapeInfo

Gör en utskrift av objektets information genom ett anrop till dess toString() metod.

Följande krav ställs på applikationen:

1. Applikationen ska utföras som en konsolapplikation där inmatning sker via menyalternativ.
2. Implementera klasserna Shape, Ellips och Rectangle enligt klassdiagrammen.
3. Egenskaper som kan sättas (har en set metod), Width och Length ska ha validering för

felaktig indata.

4. Lägg till nödvändiga metoder för menyvisning, inläsning och utskrift på programklassen.
5. Vid fel ska lättförståeliga felmeddelanden visas.

Interface

Eget interface

Utöka programmet i uppgift 1 genom att skapa en version av Rectangle som går att ändra storlek på. Detta görs genom att skapa ett interface av typen IResizable med metoden resize(int) som ska ändra storleken i procent.
Skapa sedan en klass ResizeableRectangle som implementerar interfacet IResizable.

Använda färdigt interface

Låt Rectangle och Ellipse implementera det färdiga interfacet public interface **Comparable<T>** som har en metod, compareTo.
Låt metoden göra en jämförelse om vilket objekt som har den största arean. För returvärden se dokumentationen:

https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html

Lists (and arrays) of objects that implement this interface can be sorted automatically by `Collections.sort` (and `Arrays.sort`). Objects that implement this interface can be used as keys in a sorted map or as elements in a sorted set, without the need to specify a comparator.