

SKILDPADDE 2 (TURTLE 2)

I denne opgave skal I arbejde videre med jeres skildpaddeprojekt. Denne gang skal I implementere nogle metoder, der bruger rekursivitet, dvs. kalder sig selv.

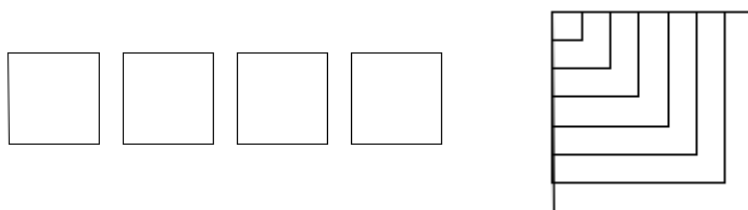
Start med at tage en kopi af jeres projekt fra sidste opgave (inklusiv de metodekroppe, som I implementerede der). Så har I altid originalen at falde tilbage på, hvis noget går helt i koks.

Husk at metoderne også skal fungere, når skildpadden starter i en vinkel, der er forskellig fra 0. De fleste af metoderne kalder **penUp** og **penDown** flere gange. Det er op til brugeren at sørge for, at pennen er nede før metoden kaldes. Hvis dette ikke er tilfældet, vil første del af figuren ikke blive vist.

Hvis I har svært ved at få figurerne korrekte, kan I med fordel indsætte **pause** metoden passende steder i jeres kode. Metoden får skildpadden til at vente 100 msek, hvilket gør det meget lettere at følge dens bevægelser. Metoden kan også kaldes med et heltal som parameter (så venter den det angivne antal msek). Metoden er omtalt under **Actor** klassen på første side af opgaveformuleringen for Skildpadde 1.

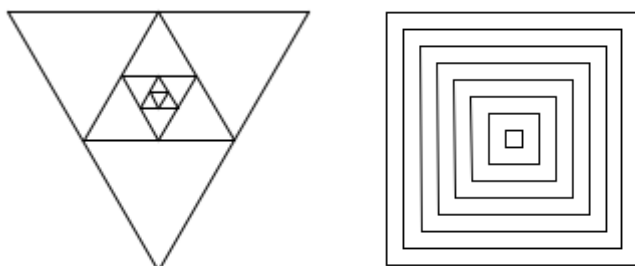
Opgave 1

Implementer metoderne **squaresHorizontal** og **squaresCornered**, således at de tegner nedenstående figurer. Hver gang der er tegnet et kvadrat, foretages et rekursivt kald (der sørger for de resterende kvadrater). For den første figur er det nemmest at starte fra venstre og så arbejde sig mod højre. For den sidste figur er det nemmest at begynde udefra og så arbejde sig indad. Det kræves ikke, at skildpadden slutter i udgangspositionen.



Opgave 2

Implementer metoderne **triangles** og **squaresCentered**, således at de tegner nedenstående figurer. Hver gang der er tegnet en trekant/kvadrat, foretages et rekursivt kald (der sørger for de resterende trekanter/kvadrater). For begge figurer er det nemmest at begynde udefra og så arbejde sig indad. Det kræves ikke, at skildpadden slutter i udgangspositionen.



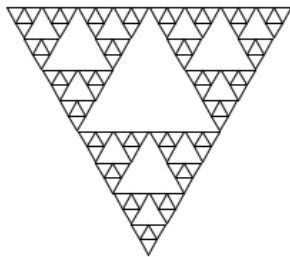
Opgave 3

Implementer metoderne **kochCurve** og **kochFlake**, således at de tegner nedenstående figurer. For den første metode bruges den fremgangsmåde, der blev beskrevet i en af forelæsningserne. Den anden metode er ikke rekursiv. Den kalder blot den første tre gange med nogle passende drej ind imellem.



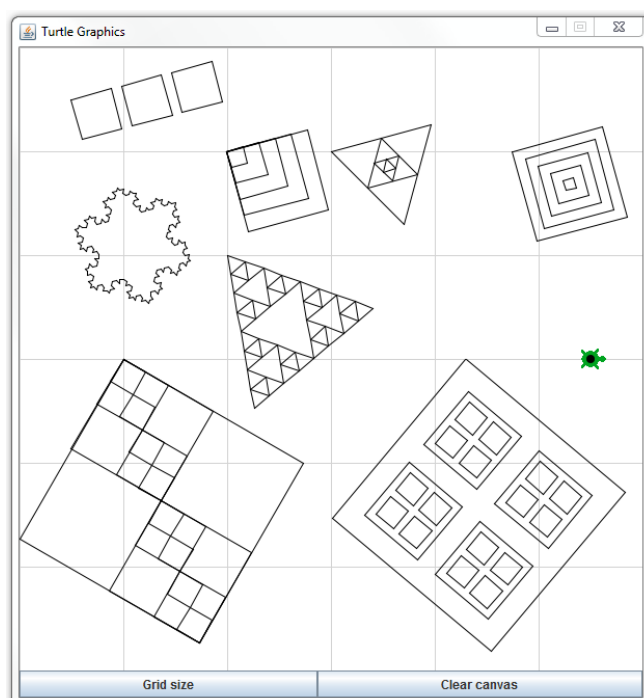
Opgave 4

Implementer metoden **sierpinskiCurve**, således at den tegner nedenstående figur. Brug den fremgangsmåde, der blev beskrevet i en af forelæsningserne. For at få det til at virke, er det vigtigt, at skildpadden i hvert af de rekursive kald slutter med at gå tilbage til udgangspositionen og indtage samme vinkel, som den havde, da kaldet startede.



Opgave 5

Skab en skildpadde og kald metoden **testTurtle2** (eller kald klassemetoden **testSecond**). Metoden skal producere nedenstående lærred (hvor de to nederste figurer først bliver tegnet, når I har løst opgave 6). Hvis jeres lærred ser forkert ud, er der noget galt med jeres metoder, og I skal derfor rette dem til. Det er selvfølgelig forbudt at modificere testmetoden.

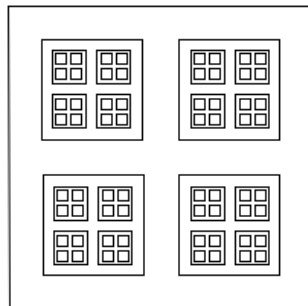
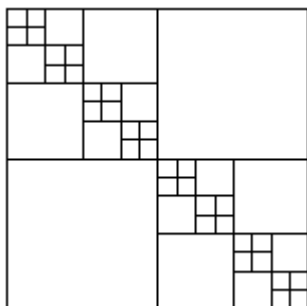


Dernæst skal I afprøve jeres program ved hjælp af testserveren. Kald klassemetoden **test** i **TestServer** klassen med parameteren "Turtle2". Hvis testserveren finder fejl, skal I gennemgå jeres kode og forsøge at rette dem.

Testserveren tjekker jeres tegninger ved at sammenligne det lærred, som I tegner på (pixel for pixel) med et lærred som nogle korrekte metoder har tegnet på. Tegnemetoderne bruger reelle tal (double), som afrundes for at kunne repræsenteres på computeren. Hvis I tegner jeres rekursive figurer på en anden måde, end vi gør, kan I derfor få et lidt andet resultat end vores og dermed et negativt svar på testserveren (idet der er nogle pixels, der ikke stemmer overens). I så fald kan I blot lave en nøje sammenligning af jeres lærred med det, der er afbildet ovenfor (læg speciel mærke til om I har det rigtige antal sider, det rigtige antal figurer inden i hinanden, den rigtige størrelse, osv.). Hvis figurerne stemmer overens kan I blot aflevere. Når I flytter rundt på Skildpadden, kan det være en fordel at bruge **jump** i stedet for **move**, således at I ikke tegner to streger oveni hinanden.

Opgave 6 (til dem, der har mod på mere)

Implementer metoderne **squarePatternDiagonal** og **squarePatternFour**, således at de tegner nedenstående figurer. For at få det til at virke, er det vigtigt, at skildpadden i hvert af de rekursive kald slutter med at gå tilbage til udgangspositionen og indtage samme vinkel, som den havde, da kaldet startede.



Afprøv de skrevne metoder ved hjælp af testserveren (som beskrevet i opgave 5).