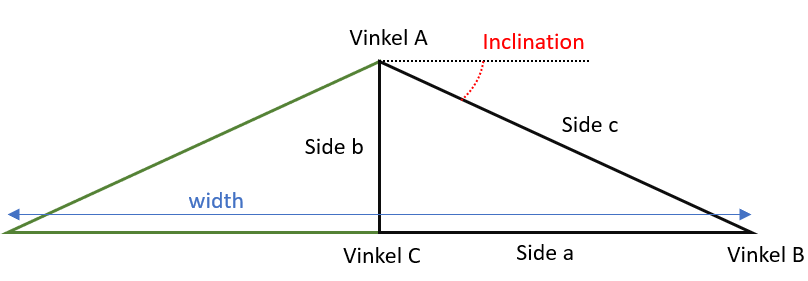
# Tests

Vi har ikke lavet mange tests i intelliJ. Vi har i stedet benyttet os af at udvikle metoderne i et separat program, for så at implementere dem direkte i klasserne når metoderne er klar, tilpasse vores variabel-navne til vores projekt og så teste om de virker ved at køre programmet. Dette var let og fantastisk hurtigt, men ikke ’best practice’ og kan give problemer i større, mere sofistikerede projekter, så vi ønsker at implementere unit testing mere i fremtiden.

Vi har dog valgt at køre et par unit tests på vores Roof klasse, da denne indeholder nogle ret ligetil, matematiske metoder. Her følger en kort forklaring:

## Forklaring

Som det ses på billedet ovenfor, så indeholder et tag mange mål. Når en kunde bygger en carport med tagrejsning, så vælger de kun hældningsgraden, hvilket vi kalder inclination. Denne grad, sammen med carportens bredde, som vi her kalder width, indføres i visse metoder som giver os de resterende mål som vi har brug for til vores stykliste.

## Siden c (calcSideC())

Siden c eller hypotenusen i vores trekant er det første vi udregner. Udover at give et godt udgangspunkt for senere at finde siden b, tagets højde, så er det en vigtig dimension, både for vores komponenter som bruges til taget eller ligger langs taget, men også direkte for vores tagbelægning.  
Der er foretaget test af maksimum og minimum værdierne for inclinationen, samt en tilfældig inclination’s værdi indenfor disse.

## Siden b (calcRoofHeight())

Siden b fungerer som tagets højde, som går fra carportens top til tagets top. Denne dimension er vigtig for vores lodrette midterspær og for beklædningen af tagets lodrette sider.  
Der er foretaget test af maksimum og minimum værdierne for side c, samt en tilfældig side c værdi indenfor disse.

## Konvertering mellem datatyper (toInt())

Vores datatyper for tal i vores projekt er altid integers, men mange af det Math. funktioner som java har, kræver double værdier(da der er behov for decimaler). Dette betyder at vi i vores to forrige metoder har været nød til at konvertere vores integers til double værdier, hvilket er let nok, men når disse værdier skal genintroduceres til programmet løber man ind i nogle problemer. Derfor har vi lavet en metode som tager imod en double værdig og returner den som nærmeste integer.  
Der er foretaget tests af værdier som rundes ned og af værdier som rundes op.