

## JAVA22, Objektorienterad programmering och Java, inlämningsuppgift 1.

På växthotellet "Greenest" kan ägare checka in sina krukväxter, för omvårdnad, medan ägarna är på semester. På hotellet blir växterna bortskämda och får sola under UV-lampor samt får optimal mängd näringsvätska av den typ som varje växt behöver.

Hotellet är specialiserat på 3 typer av växter:

- **Palmer:** En palm behöver 0,5 liter kranvatten per dag \* sin längd i meter. En palm som är 3 meter hög behöver alltså  $0,5 * 3 = 1,5$  liter vatten/dag.
- **Köttätande växter:** Dessa behöver alltid en basnivå av 0,1 liter proteindryck per dag och sedan ett extra tillskott med 0,2 liter proteindryck \* sin längd i meter. En köttätande växt som är 50 cm hög behöver alltså  $0,1 + (0,2 * 0,5) = 0,2$  liter proteindryck/dag.
- **Kaktusar:** Varje kaktus ska ha 2 cl mineralvatten/dag, oberoende av kaktusens storlek.

Eftersom formlerna är så krångliga, har hotellets ägare svårt att minnas hur mycket vätska varje växt behöver och ber er om hjälp att skriva ett litet program som hjälper till att hålla reda på hur mycket varje växt ska ha.

Systemet ska, ut ett användarperspektiv, fungera enligt följande:

En meddelande-ruta dyker upp på skärmen med texten "Vilken växt ska få mat?" och en tom rad bredvid. Ägaren fyller i växtens namn på tomma raden och programmet visar en ny ruta där det står hur många liter växten ska få samt vilken sorts vätska (kranvatten, mineralvatten eller proteindryck) som ska serveras.

Dessa växter bor på hotellet:

- Kaktusen Igge, 20 cm hög
- Palmen Laura, 5 meter hög
- Köttätande växten Meatloaf, 0,7 meter hög
- Palmen Putte, 1 meter hög

### Betygskriterier:

För att bli **Godkänd (G)** krävs att din lösning fungerar enligt ovan, det ska innehålla minst ett **arv** och minst ett **interface**, implementerade på ett relevant sätt för det ovanstående datat. Vidare måste du använda dig av **inkapsling** och någon form av **polymorfism**. Skriv en kommentar i koden där dessa förekommer så att Sigrun lätt kan se att du behärskar detta. Koden ska vara enkelt läsbar och prydligt skriven.

Vidare måste programmet demonstreras på ett professionellt sätt. För att uppnå detta, innan du demonstrerar, gå noga igenom och kontrollera att programmet verkligen fungerar. Testa att mata in samtliga växter och kontrollera att du får korrekt resultat. Mata in några felaktiga värden och kontrollera att programmet svarar på ett vettigt sätt. Du kommer att behöva visa både att programmet funkar och visa din kod. Tänk igenom vad du ska säga under redovisningen och hur du ska visa upp din kod så att du framstår som den stabila programmerare du är. För att alla ska hinna demonstrera får din demonstration ta **max 7 minuter**.

För att bli **Väl Godkänd (VG)** måste alla krav för att få G vara uppfyllda, samt att du måste visa att du behärskar **enums**(uppräknings typer) och **enhetstester**.

Skriv ett enhetstest för varje växt där du anropar varje växts uträkningsfunktion. I testet, kontrollera att din uträkningsfunktion räknat rätt genom att jämföra det värde du får ut från funktionen med det korrekta värdet för varje växt.

Din kod ska genomgående följa de Best Practices vi pratat om på lektionerna **och inga hårdkodade strängar eller siffror får förekomma.**

Koden skall vara uppladdad på GitHub i ett publikt repo, som kursledaren har åtkomst till (gäller både G och VG).

Denna uppgift görs individuellt.