

Bakteriesimulering

Notera: Denna uppgift är lite större än en tänkt duggauppgift men kanske just därför bra som övning. En annan egenskap med uppgiften är att allt som behöver göras står i texten. Följ vad som står i texten noga!

I forskningen som bedrivs inom fälten medicin och biologi är det vanligt att simulera biologiska processer. I denna uppgift ska du implementera en enkel simulering av hur bakterier växer i en koloni baserat på tillgänglig mängd näring (Notera: denna simulering är **inte** baserad på riktig forskning). För att utföra detta behöver du implementera fyra klasser:

- **Bacteria_Colony** är basklassen för en bakteriekoloni. Ett heltal representerar antalet bakterier i kolonin och ett flyttal representerar mängden näring kolonin konsumerar per steg i simuleringen. Klassen har följande funktioner:

`update()` tar emot ett flyttal som representerar tillgänglig näring. Observera att anroparens argument för detta kommer behöva modifieras av funktionen. Funktionen beräknar hur mycket kolonin växer och hur mycket näring som krävs i ett simuleringsteg enligt de regler som gäller för respektive specialiserad koloni.

`print()` tar in en utström och skriver till den information om kolonins storlek m.m. enligt det som gäller för respektive koloni. Se körexempel för respektive koloni.

- **Ecoli** representerar en koloni av Escherichia coli (E. coli) bakterier. Klassen ärver från **Bacteria_Colony** och har två datamedlemmar av flyttalstyp vid namn `growthFactor=2.0` och `growthRate=1.5`.

`update()` kontrollerar om det finns tillräckligt med näring och konsumerar i så fall $\text{delta} = \text{number of bacteria} * \text{consumption rate}$. När kolonin har konsumerat näring multipliceras antalet bakterier med `growthFactor`. `growthFactor` multipliceras sedan med `growthRate`. Om det inte fanns tillräckligt med näring sker ingenting.

- **Saureus** representerar en koloni av Staphylococcus aureus (S. aureus) bakterier. Klassen ärver från **Bacteria_Colony**.

`update()` kontrollerar om det finns tillräckligt med näring och konsumerar i så fall `delta` (se Ecoli ovan) näring samt multiplicerar antalet bakterier med 3.

- **Petri_Dish** innehåller en `std::vector` av bakteriekolonier och en datamedlem av typen `double` som representerar den totala mängden tillgänglig näring i petriskålen. Klassen har två medlemsfunktioner:

`simulate()` kör varje kolonis `update()` funktion en gång med näringen som argument.

`print()` kör varje kolonis `print()` funktion en gång och skriver till slut ut den totala mängden näring som finns kvar. Se körexempel.

Det finns ett program i filen `given_files/bacteria.cc` att bygga ut. Tänk utskrift:

```
$ ./a.out
E-coli count: 1 with growth factor: 2
S-aureus count: 1
Nutrients: 600
Simulating 5 time steps...
E-coli count: 1842 with growth factor: 15.1875
S-aureus count: 243
Nutrients: 140
```