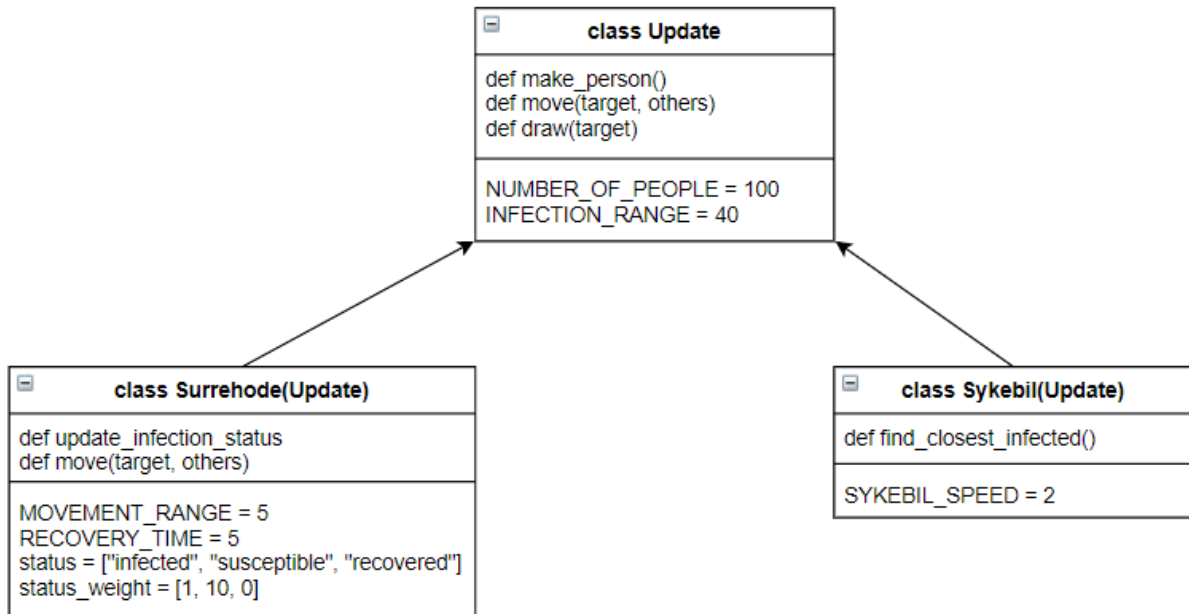


## Eksamen 2020

### Oppgave 1:

Klassediagram:



I dette eksemplet ser vi at subclassene «Surrehode» og «Sykebil» arver fra superklassen «Update». Jeg har valgt å demonstrere polymorfi mellom klasser i den form at subclassen «Surrehode» har en metode «move» med samme navn som i superklassen «Update». Grunnen er at det kan være nyttig å lage en ny funksjon for surrehodene dersom man ønsker at de skal oppføre seg på en annen måte enn sykebilene. Dersom en instans av «Surrehode» bruker en metode «move» som har samme navn som en metode i superklassen, vil metoden i subclassen bli utført.

En løsning som dette kan være bra eller dårlig avhengig av hva koden blir brukt til. For eksempel syntes jeg det er en dårlig løsning å ha funksjoner med samme navn i forskjellige sub -og superklasser da det fort kan bli vanskelig å holde styr på hvilken metode som faktisk blir utført, men på en annen side utvider man funksjonaliteten til programmet dersom man kan endre på oppførselen til objektene.

### Oppgave 2:

Oppførselen til surrehodene og sykebilene blir fra pre-koden bestemt i én funksjon. Surrehodene blir gitt en tilfeldig hastighet og vil loope gjennom veggene dersom de går for langt og sykebilene vil rette seg mot et «infected»-surrehode dersom surrehodet er innenfor en viss radius, ellers vil sykebilene stå i ro.

Siden jeg har valgt å opprette klasser for surrehoder og sykebiler mener jeg det er mer oversiktlig å implementere én oppførselsfunksjon i hver av klassene. På den måten kan man definere en spesifikk oppførsel til hver av objektene og bruke arv på den måten at variabler

som har felles betydning for oppførsel blir definert i en superklasse.

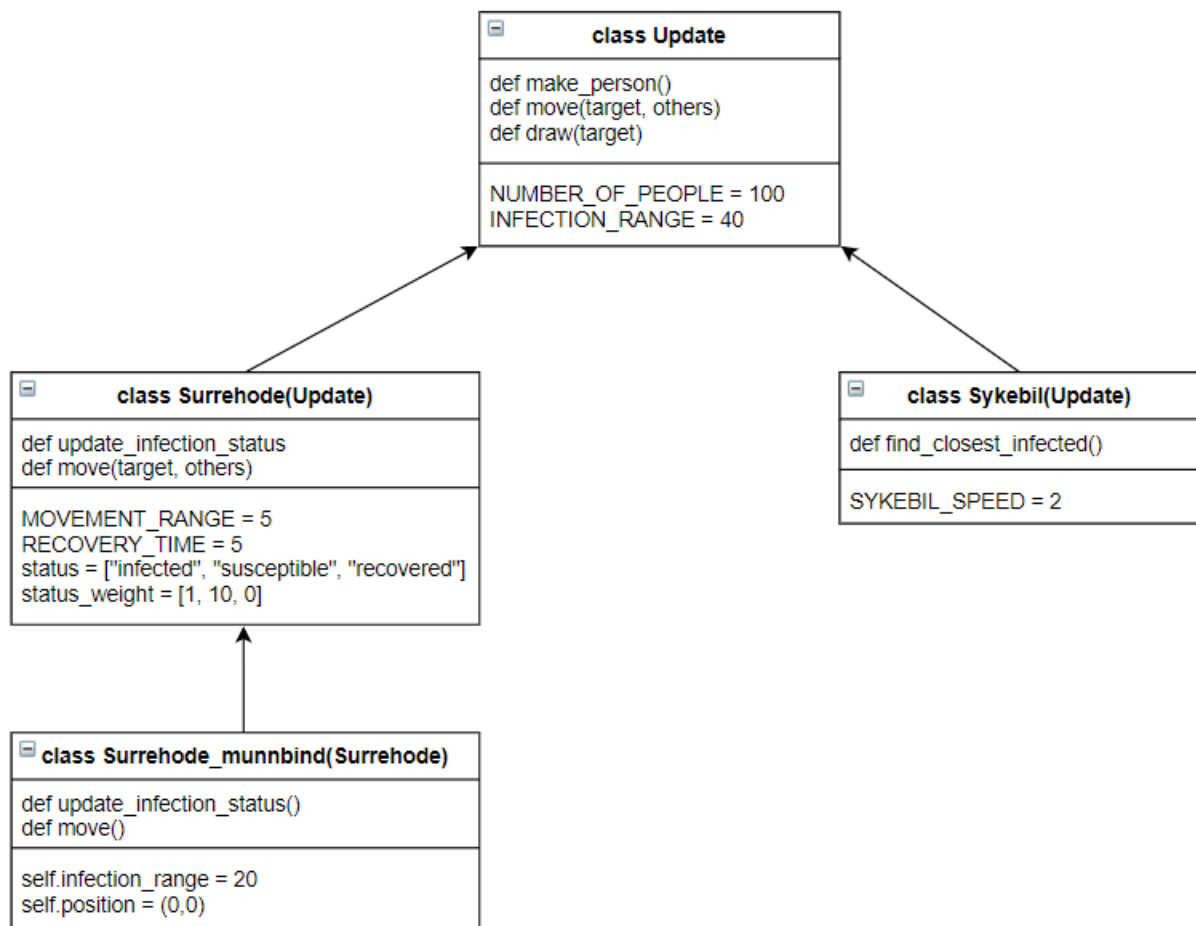
Men på denne måten kan det bli vanskelig å få demonstrert polymorfi. Derfor har jeg tenkt at det kan være en løsning å definere en oppførselsmetode i superklassen som tar en subklasse som argument. Dermed kan denne oppførselsmetoden brukes av flere klasser og subclassene kan arve fra superklassen.

### Oppgave 3:

Se vedlegget «Eksamen\_2020.py»

### Oppgave 4:

a) Jeg ville opprettet en ny klasse: «class Surrehoder\_munnbind», som er enda en subklasse av «Surrehoder». Dette fordi Surrehoder med munnbind kan ha noen attributter og metoder som er lik surrehoder uten munnbind, men også ha andre attributter og metoder som gjør at de skiller seg fra vanlige surrehoder.



b) For at dette skal virke må man legge til surrehodene med munnbind i listen «behaviours» og tildele dem en vektning i «Behaviours\_weight». På denne måten vil de også inkluderes i «make\_person» metoden. På grunn av at jeg har valgt å bruke en løsning som innebærer bruk av sprites og sprite grupper, er det også nødvendig å legge til en egen sprite gruppe for surrehodene som bruker munnbind.

I klassen for surrehoder med munnbind har jeg redusert infeksjonsradiusen slik at sannsynligheten for at de smitter andre eller blir smittet er mindre. I tillegg har jeg lagt til tre metoder som blant annet vil sjekke hvor stor prosentandel de med munnbind utgjør av totalt antall personer og flytte på surrehodene med munnbind. Metoden som flytter på surrehodene med munnbind får dem til å trekke seg unna surrehodene uten munnbind slik at smittesannsynligheten blir mindre.

For å kunne eksperimentere med sannsynligheten for at de med munnbind smitter andre eller blir smittet ville jeg implementert en metode som registrerer hvor mange personer som blir smittet over et visst tidsrom og funnet ut hvor fort viruset sprer seg. Basert på svaret vil man da kunne si noe om hvor fort man kan bli smittet og hvor stor sannsynligheten er for å bli smittet basert på hvor mange rundt en som er smittet. Jo fler som er smittet desto større er sannsynligheten for å bli smittet.

c) Se vedlegget «Eksamen\_2020.py»