

→ verschillende vaccins

- Class Vaccin (pure virtual)
 - Bv class Pfizer : public Vaccin
 - Public ? geen ensure expect nodig
 - Static float temp
 - Static int hernieuwing (tweede prik)
 - Static int interval
 - Static int transport
 - Static int levering
 - private
 - Int dag_van_vaccinatie // moet alleen als we Vaccin klassen gaan opslagen
 - Aantal // " " " " " " " "
 - We initialiseren deze variabelen in parser
- Enum VaccinType = {Pfizer, ...} // misschien ook niet nodig
- VaccinatieCentra
 - Vector<Int[3]> vaccinaties: int[3] = {VaccinType, dag_van_vaccinatie, aantal} OF
 - Vector<Vaccin> vaccinatie = {Pfizer, Pfizer, Moderna, ...}
 - +Omzetting type klasse gebeurt direct
 - -We kunnen een attribuut dagen sinds vaccinatie bijhouden in Vaccin, en dit elke dag voor elke 'batch' updaten
 - -In de andere implementatie kunnen we of de huidige dag meegeven als argument(meer verbondenheid tussen klassen dus -) of
 - - we kunnen apart de huidige dag bijhouden (meerdere variabelen in aparte klassen met verschillende functie
 - +We kunnen een klasse Simulatie aanmaken met een functie om de huidige dag op te roepen → oplossing voor beide implementaties
 - Map<VaccinType, int> buffer om vaccins te ontvangen → in nieuwe dag maken we objecten aan
 - Ontvang levering uitbreiden
 - Nieuwe dag uitbreiden
 - Iedereen gevaccineerd aanpassen
- Hub
 - Vector<Int[2] of pair> aantal vaccins: int[2] = {VaccinType, aantal}
 - Ontvang levering aanpassen (type en aantal)
 - Verdeel vaccins aanpassen // opdracht
 - – leveringen interval
- Simulatie aanpassen
 - + levering interval → checken voor elk type vaccin
- Parser
 - Nieuwe tags herkennen + nieuwe objecten aanmaken
- Output moet algemeen aangepast worden(met bijhorende getters)