

## Virussimulator



2021...De virussen blijven ons leven bepalen. Een bekend pharmabedrijf heeft jouw expertise nodig om virussen en vaccin te simuleren. Kan jij helpen?!

## Belangrijke afspraken

- Iedere klasse komt in een apart bestand.
- De volledige **solutionfolder** moet als een **zip** ingediend worden.  
*(rechterklik op je folder => Kopiëren naar "Gecomprimeerde (gezippte) map")*

## Start de klok

We maken een applicatie waarmee we vaccins en virussen gaan simuleren. In deze wereld heeft ieder virus een “killcode”: een verborgen code. Indien een vaccin de juiste killcode heeft dan kan deze gebruikt worden om een virus uit te schakelen.



### Maak een Vaccin klasse (65 punten)

Deze klasse heeft:

- Een Naam (`string`) als autoproPERTY met private SETTER.
- Enkel een overloaded constructor, waarbij je de naam van het vaccin kunt instellen.
- Een full PROPERTY `Oplossing` van het type `int` deze staat standaard op `-1` en kan nooit onder `-1` gaan.
- Een methode `TryKillCode` die geen parameters aanvaardt en steeds een random getal tussen `1` en `100` teruggeeft
  - Indien `Oplossing` een andere waarde dan `-1` heeft zal deze methode géén random getal teruggeven maar wel de waarde van `Oplossing`.
- Een methode `ToonInfo` die de naam van het vaccin en de huidige `Oplossing` op het scherm zet.
- Een static methode `DuplicateVaccin`: je kan aan deze methode een `Vaccin` object meegeven. Deze methode zal een nieuw `Vaccin` als resultaat teruggeven. Dit nieuwe `Vaccin` heeft als naam “KOPIE X” waarbij X de naam is van het vaccin dat als parameter wordt meegegeven. De `Oplossing` van het meegegeven `Vaccin` wordt ook overgenomen (ook indien deze `-1` is).



## Maak een Virus klasse (65 punten)

Deze klasse heeft:

- Een Naam als autoproperty met private setter.
- Een DoomCountdown (`int`) full property met private setter:
  - Indien doomcountdown 0 of lager wordt gezet zal er "Game over [Naam virus]" op het scherm verschijnen. Dit gebeurt in de setter van de property.
- Een private `int killcode`.
- Enkel een default constructor die:
  - DoomCountdown op een willekeurig getal tussen 10 en 20 instelt.
  - killcode op een willekeurig getal tussen 0 en 99 instelt
  - Naam wordt willekeurig als volgt: deze bestaat uit 3 willekeurige letters na elkaar, gevolgd door een getal tussen 1 en 99 (bv ABC34).
- Een methode `TryVaccin` die:
  - Eén parameter van het type `Vaccin` aanvaardt.
  - Een bool teruggeeft. Deze zal true zijn indien het meegegeven `Vaccin` werkt:
    - De methode zal de `TryKillcode` methode aanroepen op het meegegeven `Vaccin`. Indien het resultaat van `TryKillcode` overeenkomt met de killcode van het Virus zal er een true teruggestuurd worden. Ook zal vervolgens de property `Oplossing` van het `Vaccin` op de geteste én werkende killcode ingesteld worden in het meegegeven `Vaccin`.
    - Indien de kill code verkeerd is wordt er false teruggestuurd , maar niet voordat eerst de doomcountdown van het virus met 1 werd verlaagd.

## Zoeken vaccin (50 punten)

We gaan nu op zoek naar het juiste vaccin.

*Maak eerst een virus object aan.*

Maak een programma waarin je 3 vaccins aanmaakt. Vervolgens ga je deze vaccins blijven testen (in een loop) op een aangemaakt virus en toon je aan de gebruiker welke vaccins werkten. Van zodra je een werkend vaccin vindt stopt de testloop en toon je welke Vaccin de oplossing heeft gevonden. Vervolgens wordt van dit Vaccin een kopie gemaakt en toon je dat ook deze de juiste oplossing bezit door z'n naam en oplossing op het scherm te zetten.

Indien de countdown van het virus op 0 komt te staan ben je game over en heb je geen vaccin gevonden. De gebruiker zal het programma dus opnieuw moeten opstarten en hopen dat er deze keer wel een vaccin kan worden gevonden.

**Tip:** Het kan lang duren voor je een werkend Vaccin vindt. Je kan daarom overwegen om de `DoomCountdown` van je virussen tijdelijk op een erg groot getal (bv 10000) te zetten zodat je voldoende kan testen voor de countdown afloopt.

### Een voorbeeld van werking

Je maakt een virus aan dat volgende eigenschappen willekeurig heeft gekregen via z'n default constructor:

- Naam: TIP54 , killcode: 25, DoomCountdown: 12

Je maakt 3 Vaccin objecten aan met een zelfgekozen naam.

- Je roept nu telkens op het virus de methode `TryVaccin` aan en geeft afwisselend steeds een ander Vaccin door.
- Iedere keer zal in `TryVaccin` (in het virus) de "TryKillCode" methode van het meegegeven vaccin worden aangeroepen. Aangezien alle Vaccin hun oplossing op -1 hebben staan zal `TryKillCode` steeds een willekeurig getal uitproberen.
- Iedere keer dat dit gebeurt wordt de `DoomCountdown` met 1 verlaagt. Indien echter op gegeven moment de juiste killcode wordt geraden door het vaccin dan zullen, nog steeds in `TryVaccin`, er 2 zaken gebeuren:
  - De `Oplossing` property van dat Vaccin wordt verandert naar de `killCode` van het virus (vanaf nu zal dus dit Vaccin steeds dit getal als "resultaat" proberen in `TryKillCode` en niet meer een willekeurig getal)
  - Er wordt `true` teruggestuurd zodat de loop weet dat de `TryVaccin` gelukt is!