

# OIS onderzoek - Programmeer boek

## Uitleg

Voor het vak Onderzoek van Interactieve Systemen zijn wij bezig met een onderzoek over het leren van programmeren aan de hand van een AI code completion tool vergeleken met traditionele methoden.

Hieronder staat een opdracht die bestaat uit twee kleinere opdrachten. Aan jou de taak om deze opdracht te maken aan de hand van het programmeerboek. Je mag de code schrijven in Visual Studio Code. Er mag geen gebruik worden gemaakt van het internet. Maak je geen zorgen als je er niet uit komt. Wij testen niet jou als individu, maar de methode die je gebruikt.

Achteraf willen wij jou graag nog wat vragen stellen door middel van een evaluatieformulier.

Succes!

## Opdrachten

Volg bij beide opdrachten daar waar nodig het design recipe (geef een functieomschrijving, de types en doctests (minimaal 1 per functie)).

### Opdracht 1: Eén kaart uit een kaartspel

Maak een Class genaamd “Card”. In deze class moeten twee functies:

- een initializer (`__init__`) functie
- een functie genaamd ‘description’, gebruik hierbij f-string

de attributen van een Card zijn:

- de “suit”, of kleur, van de kaart (Spades, Hearts, Clubs, Diamonds)
- de “value” van de kaart (2,3, Ace, Jack, ...)

#### HINTS:

- Een voorbeeld uitkomst van de functie ‘description’ is “Ace of Spades”.
- F-string: `f"hello, {s}"`  
This will create a new string that contains hello, and then it will insert the value of the variable s. For example, if s happens to be the string "David", the f-string above would amount to "hello, David".

### Opdracht 2: Het hele kaartspel

Maak een tweede Class genaamd ‘Deck’. Deze Class bevat alleen een initializer functie. De initializer bevat alleen self als argument, omdat elke ‘Deck’ dezelfde verzameling kaarten heeft.

de initializer moet de volgende dingen bevatten:

- `self._suits`: een lijst met de vier suits zoals bij opdracht 1 aangegeven
- `self._values`: een lijst met alle values in een kaartendeck
- `self._cards` die kaartendeck genereert

#### De pseudocode voor `self._cards`:

create empty list of cards

for each suit do

    for each value do

        create new Card with that combination

        add it to the list

# OIS onderzoek - Codeium

## Uitleg

Voor het vak Onderzoek van Interactieve Systemen zijn wij bezig met een onderzoek over het leren van programmeren aan de hand van een AI code completion tool vergeleken met traditionele methoden.

Hieronder staat een opdracht die bestaat uit twee kleinere opdrachten. Aan jou de taak om deze opdracht te maken aan de hand van Codeium. Je mag de code schrijven in Visual Studio Code. Er mag geen gebruik worden gemaakt van het internet. Maak je geen zorgen als je er niet uit komt. Wij testen niet jou als individu, maar de methode die je gebruikt.

Achteraf willen wij jou graag nog wat vragen stellen door middel van een evaluatieformulier.

Succes!

## Opdrachten

Volg bij beide opdrachten daar waar nodig het design recipe (geef een functieomschrijving, de types en doctests (minimaal 1 per functie)).

### Opdracht 1: Eén kaart uit een kaartspel

Maak een Class genaamd “Card”. In deze class moeten twee functies:

- een initializer (`__init__`) functie
- een functie genaamd ‘description’, gebruik hierbij f-string

de attributen van een Card zijn:

- de “suit”, of kleur, van de kaart (Spades, Hearts, Clubs, Diamonds)
- de “value” van de kaart (2,3, Ace, Jack, ...)

#### HINTS:

- Een voorbeeld uitkomst van de functie ‘description’ is “Ace of Spades”.
- F-string: `f"hello, {s}"`  
This will create a new string that contains hello, and then it will insert the value of the variable s. For example, if s happens to be the string "David", the f-string above would amount to "hello, David".

### Opdracht 2: Het hele kaartspel

Maak een tweede Class genaamd ‘Deck’. Deze Class bevat alleen een initializer functie. De initializer bevat alleen self als argument, omdat elke ‘Deck’ dezelfde verzameling kaarten heeft.

de initializer moet de volgende dingen bevatten:

- `self._suits`: een lijst met de vier suits zoals bij opdracht 1 aangegeven
- `self._values`: een lijst met alle values in een kaartendeck
- `self._cards` die kaartendeck genereert

#### De pseudocode voor `self._cards`:

create empty list of cards

for each suit do

    for each value do

        create new Card with that combination

        add it to the list