

1. Stack en heap

2. Geheugenmanagement, uitzonderingen en namespaces



Geheugengebruik in .NET

Stacks vs heaps

Introductie

- Geheugengebruik in .NET
- Twee soorten geheugen: **stack** en **heap**



Stack vs heap

- Het kleine, maar snelle stack geheugen
- Het grote, maar tragere heap geheugen

STACK

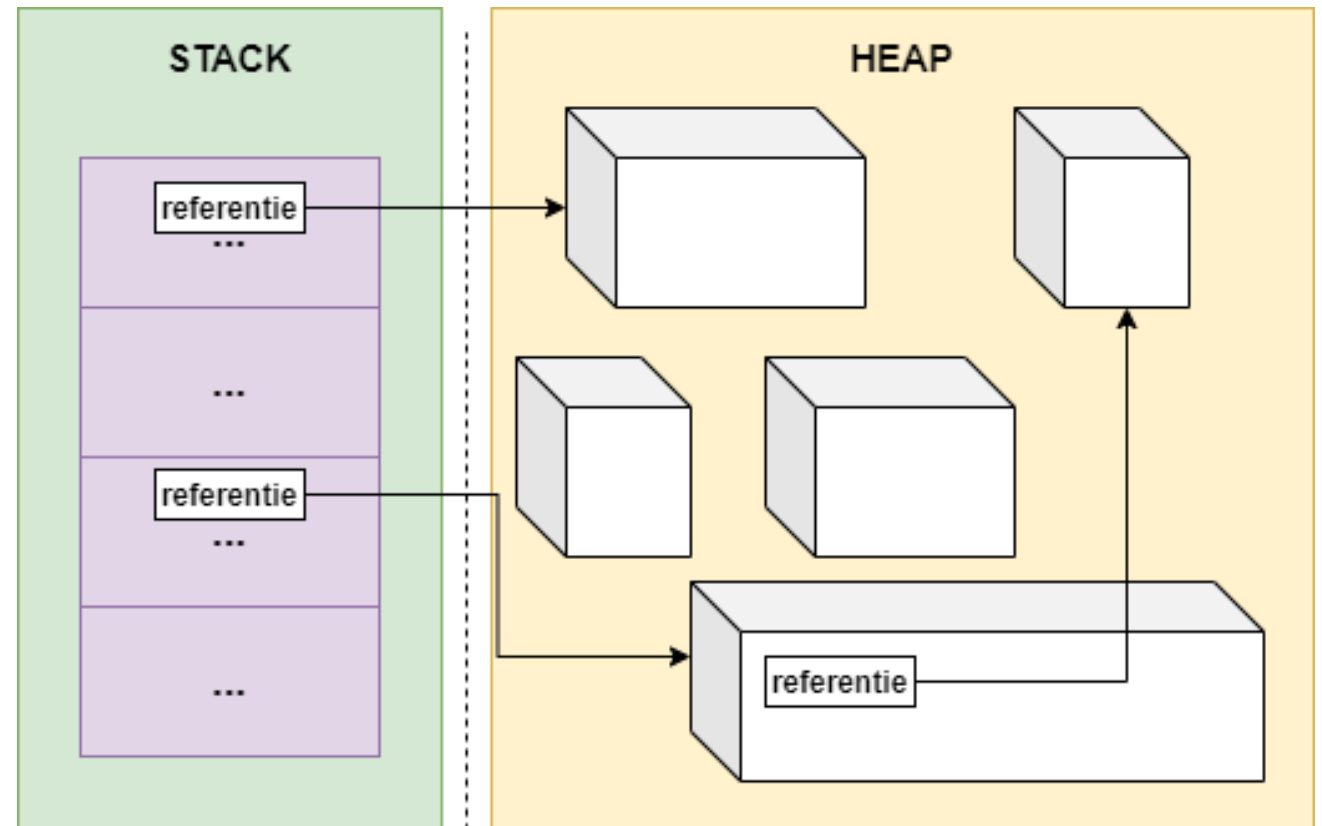


HEAP



Stack vs heap

- Value types (int, double, char, short, etc) gaan in stack
- Referenties/Pointers idem
- Objecten in heap



Wat gaat waar?



	Value types	Reference types
Inhoud van de variabele	De eigenlijke data	Een referentie naar de eigenlijke data
Locatie	(Data) Stack	Heap (globaal)geheugen
Beginwaarde	<code>0</code> , <code>0.0</code> , <code>""</code> , <code>false</code> , etc.	<code>null</code>
Effect van = operator	Kopieert de actuele waarde	Kopieert het adres naar de actuele waarde

Werking heap/stack bij reference types

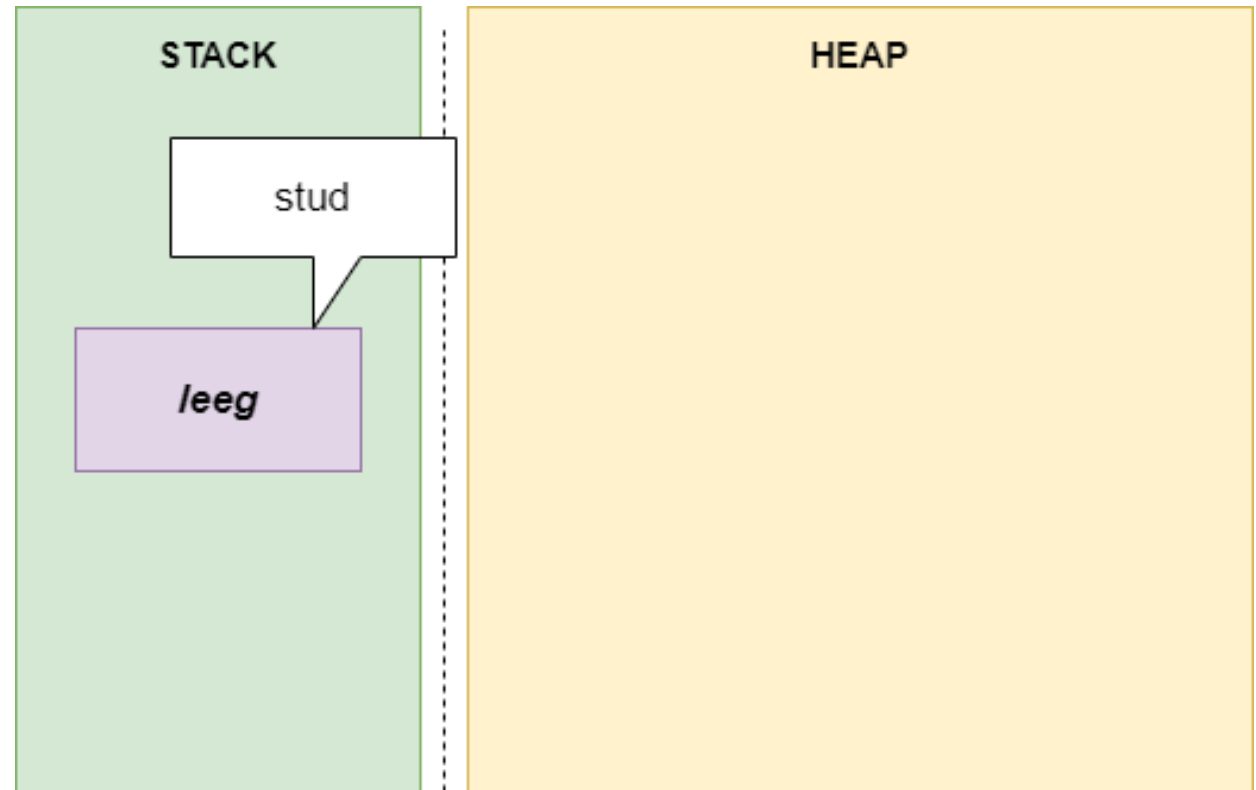
```
Student stud;  
stud = new Student();
```

STACK

HEAP

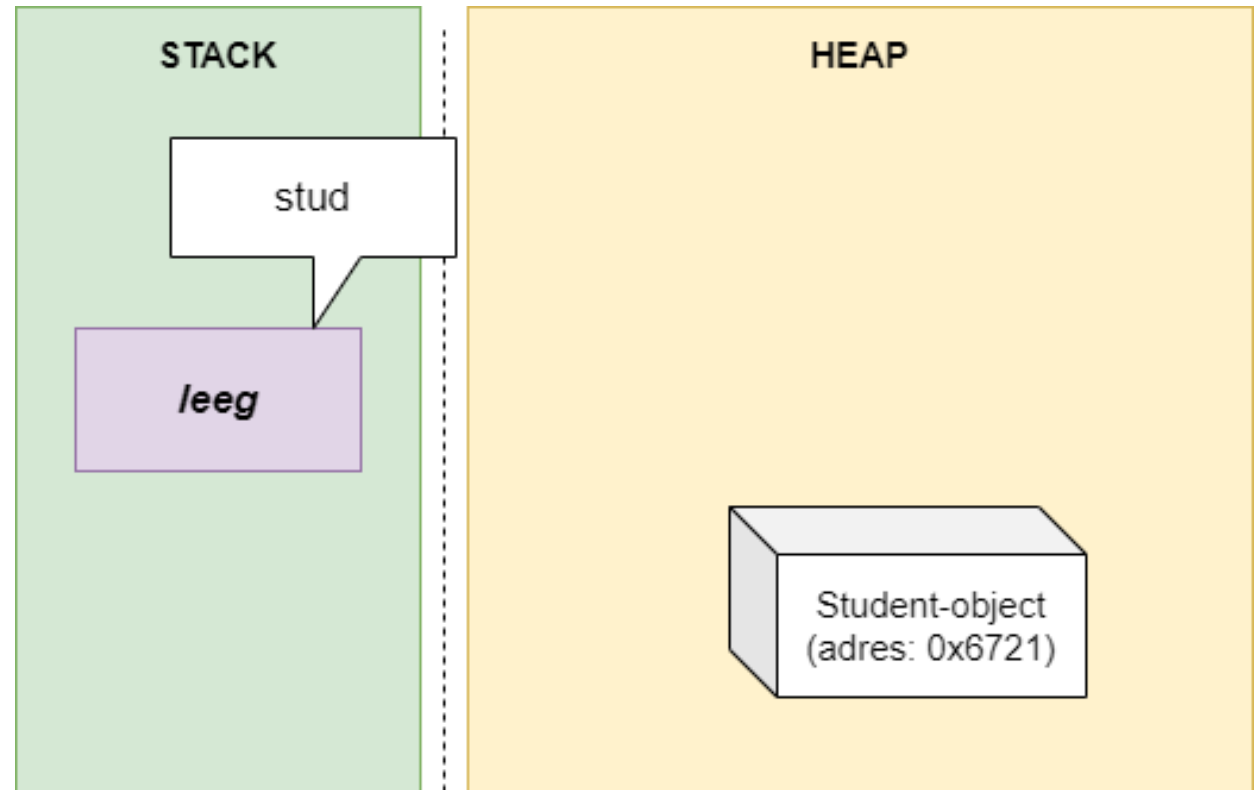
Werking heap/stack bij reference types

```
Student stud;  
stud = new Student();
```



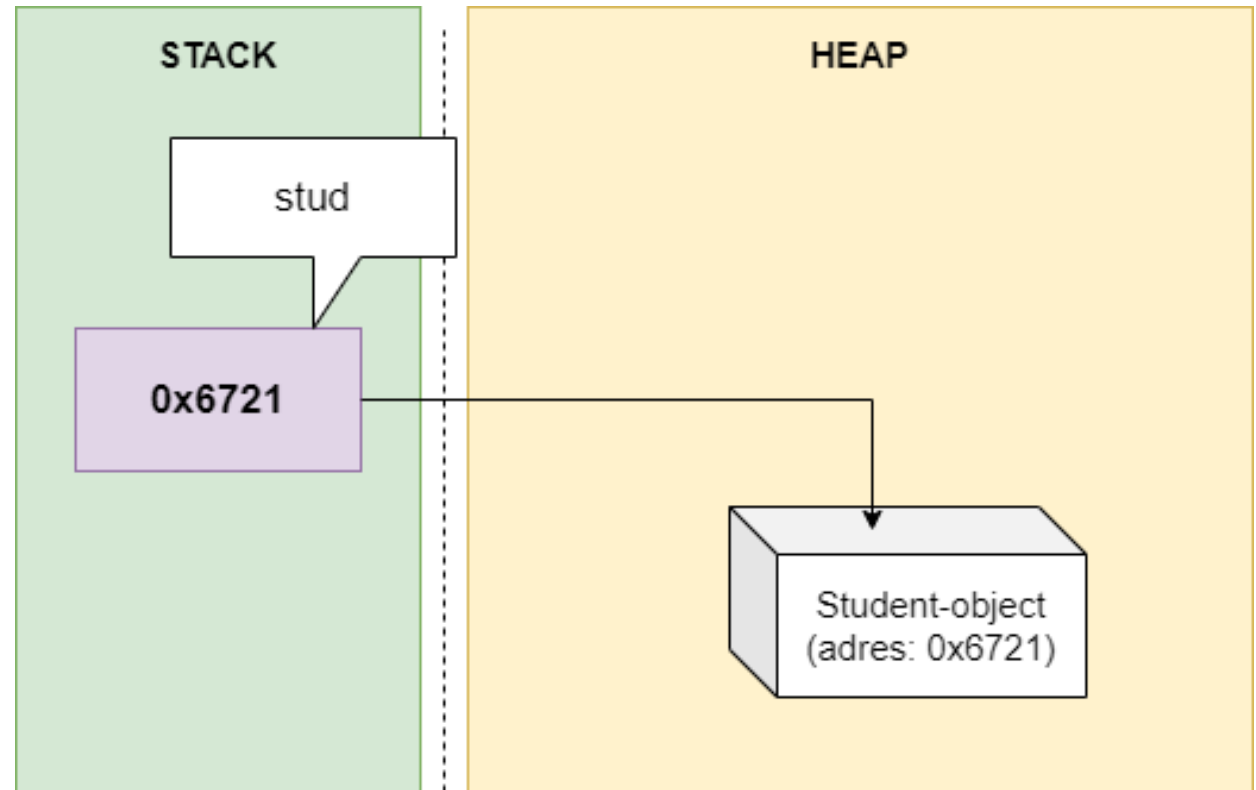
Werking heap/stack bij reference types

```
Student stud;  
stud = new Student();
```



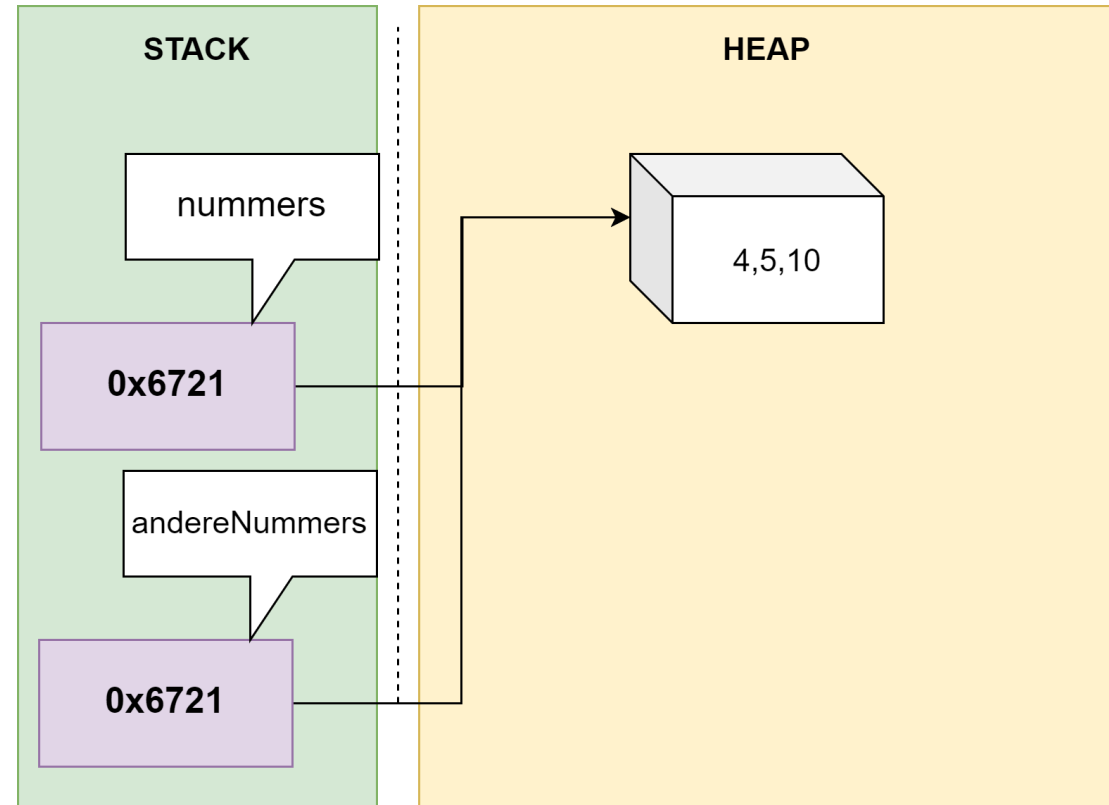
Werking heap/stack bij reference types

```
Student stud;  
stud = new Student();
```



Ook arrays zijn reference types

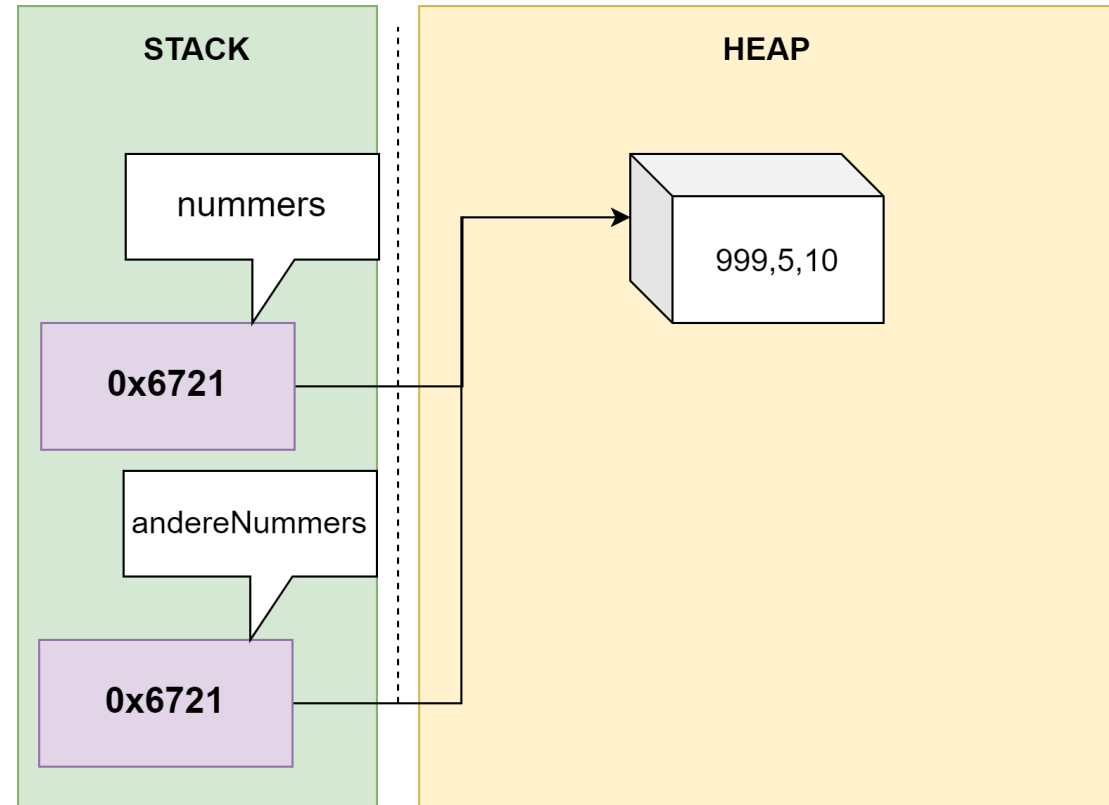
```
int[] nummers = {4,5,10};  
int[] andereNummers = nummers;
```



Ook arrays zijn reference types

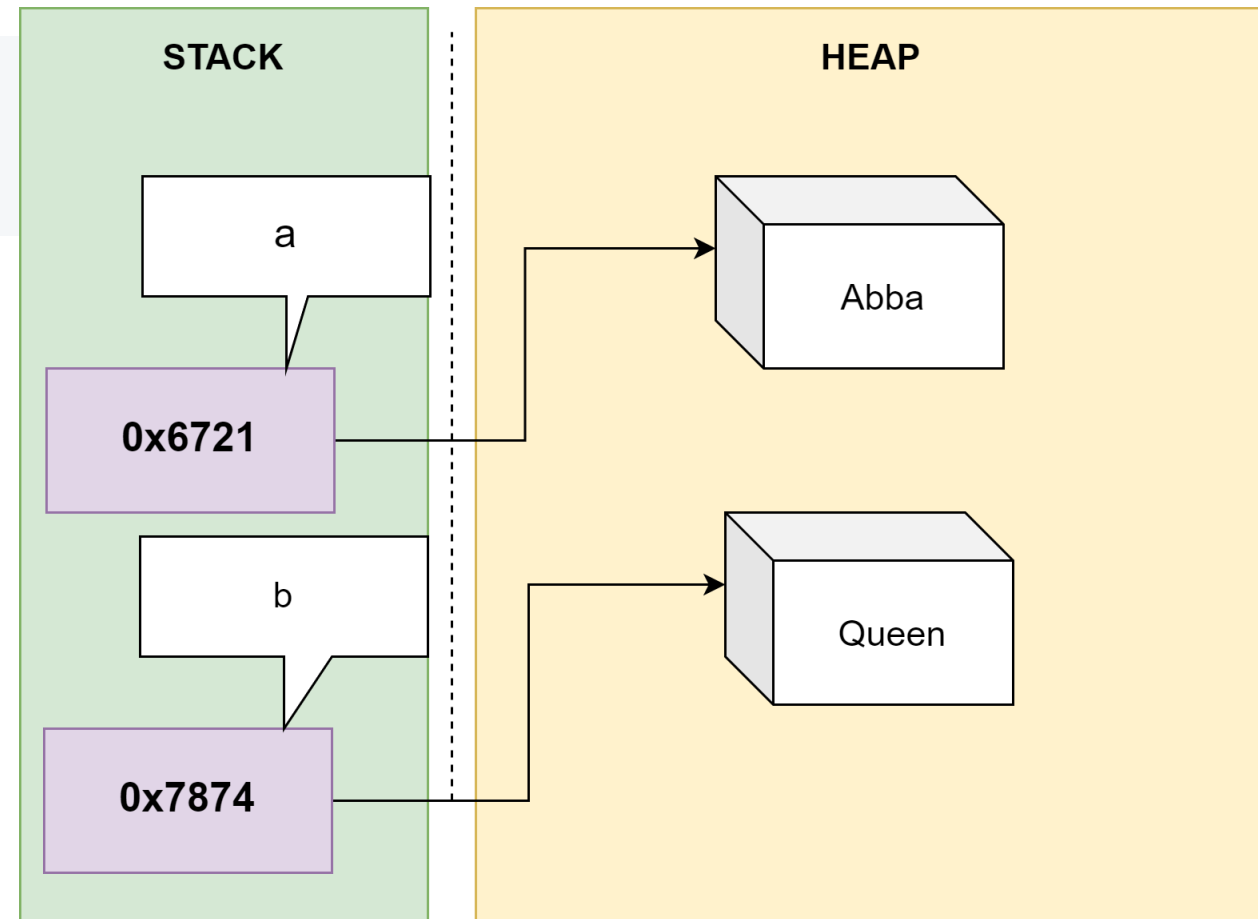
```
int[] nummers = {4,5,10};  
int[] andereNummers = nummers;
```

```
andereNummers[0] = 999;
```



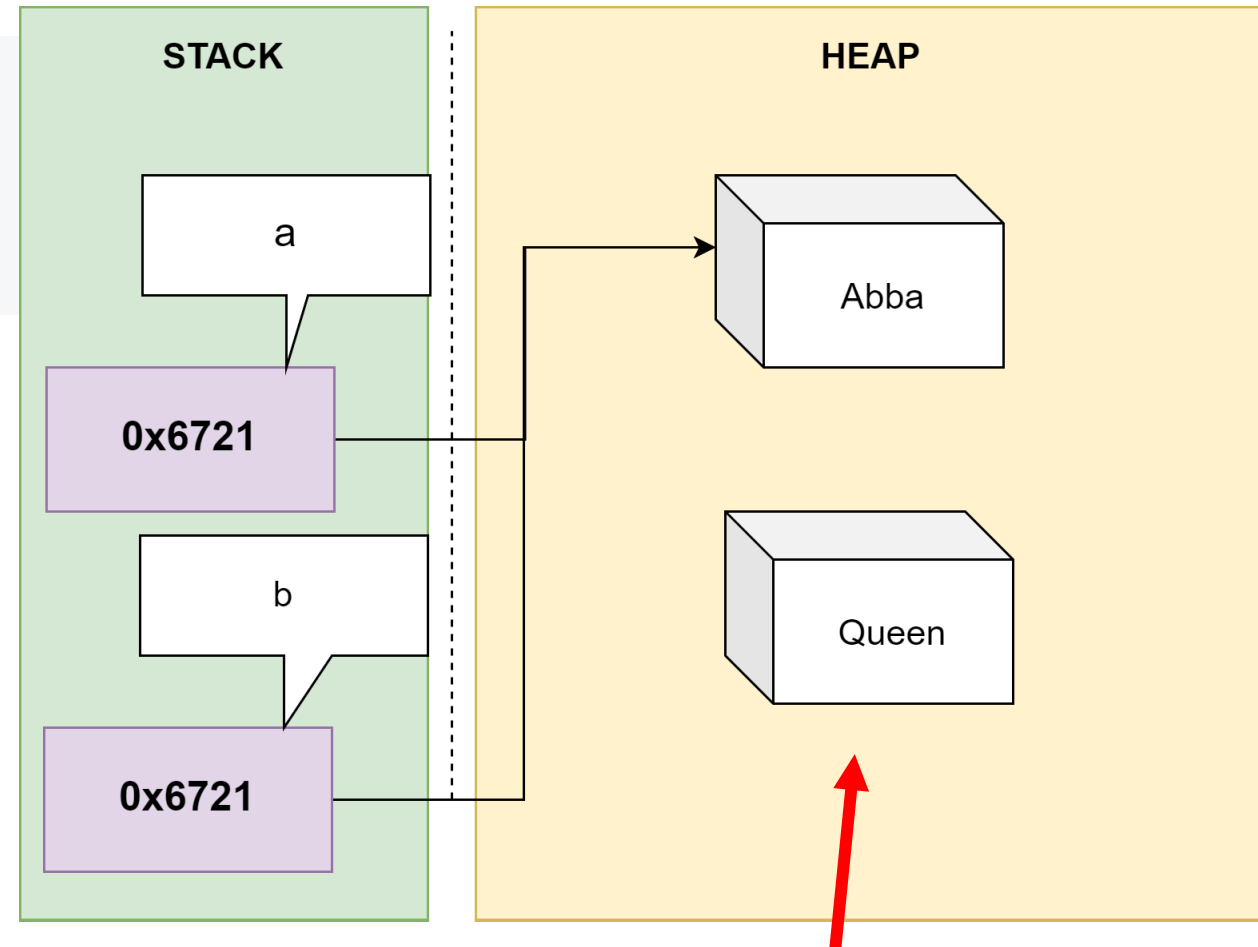
Idem bij objecten

```
Student a = new Student("Abba");
Student b = new Student("Queen");
```



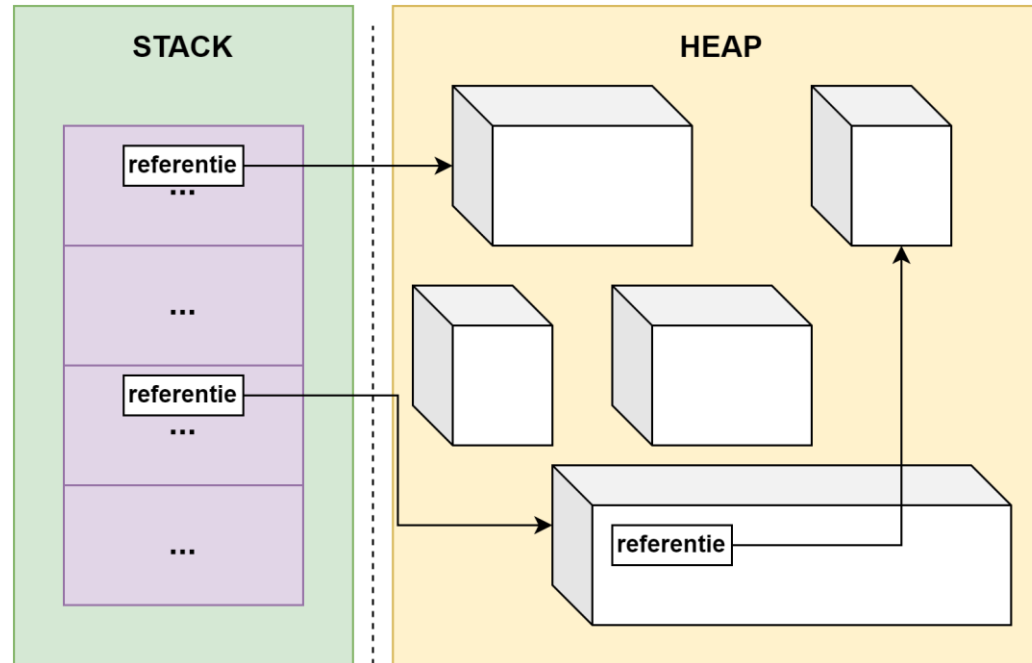
Idem bij objecten

```
Student a = new Student("Abba");
Student b = new Student("Queen");
b = a;
```



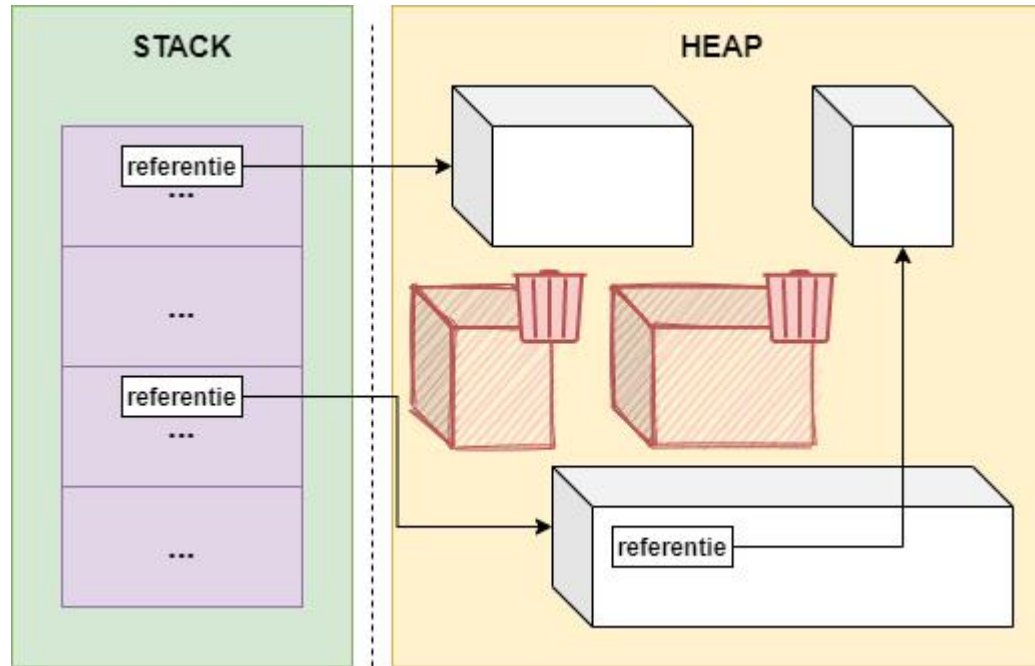
Maar wat met dit “orphan object” ???

Garbage collector (GC) verwijderd ieder object in heap waar niemand naar **refereert**.



Werking heap met garbage collector (.NET)

Garbage collector (GC) verwijderd ieder object in heap waar niemand naar **refereert**.



Werking heap met garbage collector (.NET)

Garbage collector (GC) verwijderd ieder object in heap waar niemand naar **refereert**.

