

Cocoa Objecten Definiëren

23 april 2005, Eindhoven

© Patrick Machielse

patrick@hieper.nl

Overzicht

- @interface en @implementation
- .h en .m files
- #import
- init
- dealloc

Een nieuwe Class definiëren

```
// de declaraties
@interface ClassNaam : SuperClassNaam
{
    // ivar declaraties
}
// class methoden declaraties
// instance methoden declaraties
@end

// de feitelijke code
@implementation ClassNaam
// class methoden implementaties
// instance methoden implementaties
@end
```

Een nieuwe Class (Voorbeeld)

```
@interface Cirkel : NSObject
{
    float diameter;
    NSColor *kleur;
}
- (float)diameter;
- (NSColor *)kleur;
@end

@implementation Cirkel
- (float)diameter {
    return diameter;
}
- (NSColor *)kleur {
    return kleur;
}
@end
```

Splitsen van Header en Implementation

Cirkel.h

```
("header file")

#import <Cocoa/Cocoa.h>

@interface Cirkel : NSObject
{
    float diameter;
    NSColor *kleur;
}
- (float)diameter;
- (NSColor *)kleur;
@end
```

Cirkel.m

```
("implementation")

#import "Cirkel.h"

@implementation Cirkel
- (float)diameter {
    return diameter;
}
- (NSColor *)kleur {
    return kleur;
}
@end
```

self, super

```
// Het keyword self verwijst naar het huidige
// object. B.v. de -omtrek methode van Cirkel:
- (float)omtrek {
    return [self diameter] * 3.14;
}

// super verwijst naar de super class
// B.v. de -diameter methode van de Cirkel sub
// class GrotereCirkel:
- (float)diameter {
    return [super diameter] + 1.0;
}
```

Object-initialisatie

```
// tijdens het maken van een nieuwe instance  
// wordt altijd een -init methode aangeroepen
```

```
- (id)init  
{  
    // laat eerst super class initialiseren  
    // als dat succesvol is, stel diameter in  
    self = [super init];  
    if ( self ) {  
        diameter = 1.0;  
        kleur = [[NSColor redColor] retain];  
    }  
    return self;  
}
```

Object-vernietiging

```
// dealloc wordt automatisch aangeroepen als  
// de retain count van een object 0 is
```

```
- (void)dealloc  
{  
    // eigen objecten 'releasen'  
    // niet nodig voor char, int, float, double ivars  
    [kleur release];  
  
    // laat superclass geheugen vrij maken  
    [super dealloc];  
}
```


Demo

- Definiëer de Cirkel class
- Print description