

Klassen en eigenschappen.

- **commentaar**: end -- commentaar op het einde
 - heeft geen betekenis in Eiffel, enkel ter uitleg voor ontwikkelaars.
- klassentekst **begint met class** gevolgd door de naam van de klasse
 - geen onderscheid tussen kleine en hoofdletters
- klassentekst **eindigt met end** - - class naam
- **creation-clausule** : geeft aan hoe objecten aangemaakt kunnen worden.
Hier worden de featurers (eigenschappen) opgesomd
(een voor de hand liggende naam voor een creatieprocedure is maken)
- 3 soorten features:
 - attributen = objecten die men moet onthouden, kunnen variabel of constant zijn.
 - procedures, deze wijzigen de attributen
 - functies, gebruiken attributen om bepaald resultaat te maken
- Eiffel beschikt over basisbibliotheek met volledige klassenbeschrijvingen. Deze kunnen gebruikt worden in nieuwe klassenbeschrijvingen. :
 - Integer (geheel getal)
 - Boolean (true/false)
 - Real (reële getallen met enkelvoudige precisie)
 - Double (reële getallen met dubbele precisie)
 - Character (enkelvoudige karakters)
 - String (een stuk tekst, dit is eigenlijk een samengestelde klasse)
- Het declareren van variabelen:
 - titel : STRING
 - meerdere definities binnen 1 declaratie via komma's m,n: INTEGER
- constanten worden gedeclareerd door een is-clausule i: INTEGER is 3
- Het declareren van procedures:
 - feature** procedurenaam is
 - do**
 - instructies
 - end** -- procedurenaam
- hierbij kan men parameters toevoegen als bijkomende gegevens element
feature procedurenaam (parm_1: CLASS)
- lokale werkattributen
 - feature
 - local

-- lokale attributen voor deze procedure

- Het declareren van functies
- feature
- local
- do -- instructies in de functie
- Result:=...
- end --

Invoer met “io” (io is een attribuut , kan vb scanner zijn dan scanner.readline)

- readint: lezen van integer
 - lastint = waarde van laatst gelezen integer
- readreal: lezen van een reël getal
 - lastreal = waarde van laatst gelezen real
- readchar : lezen van karakters
- lastchar = waarde van laasts gelezen char
- readline = lezen van een reeks karakters
 - laststring = waarde van de laatst gelezen reeks karakters

opdracht : io.readline

- putint (some_integer) : schrijven van geheel getal
- putreal (some_real)
- putchar (some_char)
- putstring (some_string)
- new_line

let op ! io.putstring(my_string) → mieke
io.putstring(“my_string”) → my_string

- Normaal steeds coordinator die elk object creëert en coördineert wat wanneer moet gebeuren.

- Let op:

%N → nieuwe lijn
%% → %
%T → tab
%S → #
%A → @
%” → “
%’ → ‘
%(→ (
%) →)

$$\begin{aligned} \%< &\rightarrow \{ \\ \%> &\rightarrow \} \end{aligned}$$