GROEIPROJECT
MODULE 3: REFLECTION EN ANNOTATIONS

1. Voorbereiding

1.1. Open het groeiproject en maak daarin een derde module: `3_reflection`

1.2. Kopieer ALLEEN de package **`model`** uit module `1` (met o.a. de basis- en de multiklasse) naar de nieuwe module.

1.3. Creëer een nieuwe klasse die als superklasse voor de basisklasse zal dienen (Dit heet "*generalisatie*"). In ons voorbeeld zal `Dictator` overerven van `Persoon`. Verplaats een aantal attributen van de basisklasse naar de superklasse. Verplaats ook de getters/setters die bij deze attributen horen. In ons voorbeeld verschuiven de attributen `naam`, `geslacht` en `geboorte` van `Dictator` naar `Persoon`. Zorg ervoor dat het min of meer conceptueel klopt.
→ Maak voor deze stap zeker gebruik van de menukeuze "*Refactor* > *Extract* > *Superclass*".

2. Reflection

2.1. Maak onder een nieuwe package `reflection` de klasse `ReflectionTools`. Maak daarin de volgende methode:

```
public static void classAnalysis(Class aClass)
```

2.2. De bedoeling is dat je in deze methode de klasse analyseert die als parameter binnenkomt (`aClass`). Opgelet: je moet alles kunnen doen met deze `aClass`-parameter; je mag dus nergens een cast doen naar een andere klasse!

2.3. Doe een oplijsting van de volgende informatie over de klasse en druk af in een keurig overzicht (zie ook afdruk op de volgende pagina) Overgeërfde informatie hoeft niet getoond te worden.

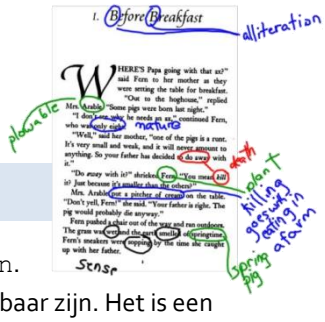
a. volledige klassenaam ("fully qualified name")
b. naam van de superklasse
c. naam van de package
d. gebruikte interfaces
e. aanwezige constructors
f. namen van private attributen
g. getters, setters en andere methoden

2.4. Maak een main-klasse `Demo_3` waarin je de methode `classAnalysis` oproept met als parameter je basisklasse en vervolgens met als parameter je multiklasse.

2.5. Gewenste output:

```
Analyse van de klasse: Dictator
================================================
Fully qualified name   : be.kdg.model.Dictator
Naam van de superklasse: Persoon
Naam van de package    : package be.kdg.model
Interfaces: Comparable
Constructors:
       public be.kdg.model.Dictator()
       public be.kdg.model.Dictator(java.lang.String,be.kdg.model.Geslacht,
       java.time.LocalDate,java.lang.String,java.lang.String,int,double,double)
attributen:    : land regime duur wreedheid slachtoffers
getters        : getRegime getWreedheid getSlachtoffers getDuur getLand
setters        : setRegime setWreedheid setSlachtoffers setLand setDuur
andere methoden: toString

Analyse van de klasse: Dictators
================================================
Fully qualified name   : be.kdg.model.Dictators
Naam van de superklasse: Object
Naam van de package    : package be.kdg.model
Interfaces:
Constructors:
       public be.kdg.model.Dictators()
attributen:    : dictatorSet
getters        :
setters        :
andere methoden: toString verwijder gesorteerdOpNaam gesorteerdOpGeboorte
gesorteerdOpWreedheid voegToe
```



3. Annotations

3.1. Maak onder de package `reflection` een nieuwe annotation met de naam `@CanRun`. Deze annotation zal gebruikt worden boven een methode en moet at runtime beschikbaar zijn. Het is een zogenaamde "marker" annotation, dus zonder parameters.

3.2. Gebruik deze annotation in je basisklasse en de superklasse daarvan (in ons voorbeeld `Dictator` en `Persoon`) boven een aantal methoden (bijvoorbeeld de setters die een String als parameter hebben).

3.3. Schrijf in de klasse `ReflectionTools` een nieuwe static methode `runAnnotated` met volgende signatuur:

```
public static Object runAnnotated (Class aClass)
```

3.4. In deze methode moet je weer vertrekken vanuit de `aClass`-parameter; je mag dus nergens een cast doen naar een andere klasse!

a. Maak een nieuwe instantie aan van deze klasse; roep daartoe de default constructor aan.
b. Overloop de methoden van deze klasse en selecteer enkel de methoden die geannotteerd zijn via `@CanRun`.
c. Voer deze methoden uit met als parameter de String "dummy".
d. Retourneer het gecreëerde object.

3.5. Roep in de main-klasse `Demo_3` de methode `runAnnotated` op met als parameter je basisklasse (in ons voorbeeld: `Dictator.class`). Druk het ontvangen Object af.
Mogelijke output:

```
Aangemaakt object door runAnnotated:
dummy              (°2000) dummy          regime: dummy              0,0 mln doden
```