Code conventies

De code is in het Engels. We houden de conventies van Java aan. De Java code conventies zijn als volgt:

1. Bestands volgorde
   * Packages
   * Import
   * Class / interface definities
2. Class / Interface
   * Classes worden geschreven in PascalCase (bv. MyClass, DataModel)
   * Interfaces gebruiken bijvoeglijke naamwoorden (bv. Serializable, Runnable)
   * Interfaces hoeven niet met een I voor de naam. In IntelliJ IDE zie je een symbool voor de class dat aanduidt wat voor type class het is
3. Indentation en Braces

Afbeelding met schermopname, tekst, Lettertype

Automatisch gegenereerde beschrijving

1. Namen van variabelen
   * Variabelen en methodes gebruiken camelCase (bv. myVariable, calculateTotal())
   * Constants gebruiken UPPERCASE\_SNAKE\_CASE (bv. MAX\_VALUE, DEFAULT\_TIMEOUT)
2. Methodes
   * De naam van methodes moeten beschrijvend zijn in camelCase (bv. calculateTotalAmount()).
   * Parameters moeten ook in camelCase
3. Comments
   * Javadoc comments voor code documentatie voor classes, methodes en properties
   * Inline comments voor kleine beschrijvingen. Geen lange inline comments waarin overduidelijke dingen in staan
4. Witruimte
   * Overal spaties tussen ( bv. Int total = 10 + 20; )
5. Error handling
   * Niet zomaar exceptions of lege catch statements gooien
   * Gebruik betekenisvolle error berichten
6. Naam conventies
   * Packages: altijd kleine letters
   * Classes / Interfaces: altijd zelfstandige naamwoorden
   * Boolean: gebruik voorvoegsels zoals ‘is’, ‘has’ of ‘can’ (bv. isValid, hasAccess)

### DTO

DTO’s kunnen op verschillende manieren gebruikt worden. We maken onderscheid in:

* API Endpoint DTO
* Persistentie DTO

Voor de API hanteren we ‘request’ of ‘response’. Een voorbeeld voor het beheren van gebruikers: ‘GetUserRequest’ of ‘GetUserResponse’. Dit is de REPR-pattern. ‘Request EndPoint Response’

Voor domein modellen die je om moet zetten naar een data object gebruik je ‘Dto’ als achtervoegsel. Dus bijvoorbeeld ‘userDto’