Objectgericht Programmeren

Coding Rules

1. Gebruik een **zelfstandig naamwoord** of een aaneensluiting van zelfstandige naamwoorden voor namen van een **klasse**. Gebruik **Pascal**-notatie voor de spelling:
   1. Vb: *BankAccount; GoldCard; PaperBackBook; …*
2. Klasse-documentatie commentaar begint met een **opsomming** van alle **eigenschappen** van de **klasse** zelf zowel als de eigenschappen van de individuele objecten binnen de klasse
3. Voeg altijd minstens een **versie-nummer** en een lijst van **auteurs** in het begin van de klasse.
4. Gebruik **werkwoorden** voor **methodes** en **camel casing.**
   1. VB: *transferTo; setBlocked; hasHigherBalanceThan*
5. Kies **namen** van **methodes** zo, zodat ze te lezen zijn als **commando’s** of **vragen** in natuurlijke taal.
6. Begin de **definities** van een klasse met een **definitie** van al zijn **constructoren**.
7. **Groepeer** alle elementen van een klasse **volgens** een bepaalde **eigenschap**. Eerst **publieke** elementen, daarna **private** elementen.
8. Maak “**getters**” en “**setters**” voor alle aanpasbare variabelen. Maak enkel getters voor niet aanpasbare variabelen.
9. Methodes mogen **niet** tegelijkertijd een object **aanpassen** en een resultaat **teruggeven**.
10. Inspectoren (getters) passen nooit een object aan.
11. Mutatoren (setters) geven nooit een resultaat terug.
12. **Inspectoren** staan **voor** **mutatoren** als ze dezelfde toegangsrechten hebben (public/private) en behoren dezelfde eigenschap.
    1. Vb:
       1. Inspectoren:
          1. getBalance
          2. hasHigherBalanceThan
       2. mutatoren:
          1. withdraw
          2. deposit
          3. transferTo
13. Gebruik **instantiatie** methodes in plaats van **statische** methodes als mogelijk.
14. Gebruik **statische** methodes als alleen **primitieve** waardes gebruikt worden.
15. Als het **gedrag** van methodes voor **alle** **objecten** van de klasse het **zelfde** is, moet je **klasse**-**methodes** gebruiken.
16. Gebruik **zelfstandige** **naamwoorden** voor namen van **formele** **argumenten** en **lokale** **variabelen**. Gebruik **camel** **casing**.
17. Start documentatie commentaar voor methodes met een **één-lijnige** opsomming van de semantiek van de methodes.
18. Verduidelijk in een methode zijn documentatie de rol van elk formeel argument met @param xyz
19. Maak basis inspectoren en voeg @Basic toe.
20. Voeg @Immutable toe aan inspectoren die onveranderlijk zijn