

Beoordeling SOPX3 project

De deelbeoordelingen zijn in een 4 punts schaal. Deze schaal is als volg:

Deelcijfer	Betekenis
A	Heel goed (≥ 8)
B	Voldoende/Goed (≥ 6 && < 8)
C	Onvoldoende (≥ 4 && < 6)
D	Zeer slecht of niet gedaan

Beoordeling per groep:

Beoordeling	1	2
Eisen + Use-cases (week 1)	B	C
Klassediagram + (week 2)	A	B
Sequencediagram + Toestandsdiagram + eerste versie (week 3)	A	B
2 ^{de} versie eisen + use-cases + aangeven wat al klaar is	-	D
2 ^{de} versie klassediagram	-	C
2 ^{de} versie sequencediagrammen en toestandsdiagrammen	-	C
Implementatie	-	C
Toepassen Libraries (STL en/of VCL)	-	B
Algemene indruk tweede versie	-	B
Eindbeoordeling	3	7

Opmerkingen per groep:

Groep 1:

- UML diagrammen
 -
- Implementatie
 -

Groep 2:

- UML diagrammen
 - In class diagrammen ontbreken namen (beschrijvingen) bij associaties.
 - Zeer uitgebreide sequencediagrammen. Deze sequencediagrammen zijn echter vanuit de code automatisch aangemaakt. Erg verhelderend zijn ze dus niet. Sequencediagrammen op een hoger abstractieniveau, die tijdens de ontwikkeling van de software gebruikt zijn, ontbreken.
 - Er is een Activiteitendiagram getekend maar de relatie tot de andere diagrammen is mij niet duidelijk.
 - Gebruik van TThread wordt verder niet toegelicht. Het is onduidelijk wat de taak van de verschillende threads is en hoe ze samenwerken.
 - Er is niet aangegeven wat al klaar is. De tweede versie van het use-casediagram is hetzelfde als de eerste versie.
- Implementatie
 - In Centraalband::MoetontvangDoos wordt een Simulatie *sim aangemaakt die niet wordt geïnitialiseerd maar wel gebruikt wordt. Als deze code uitgevoerd wordt zal een general protection error optreden!
 - Idem in Centraalband::schuifBand.
 - In Centraalband::schuifBand wordt een new Wachten aangemaakt die nooit meer gedelete wordt. Dit levert een memory leak op.

- In class SortUitBuf worden 2 int variabelen gebruikt als semafoor. Dit zorgt ervoor dat busy waiting gebruikt wordt. Het werkt niet correct omdat de test en set van de variabele niet ondeelbaar is. Om correct te synchroniseren kun je TCriticalSection van Borland gebruiken.
- In Centraalband::schuifBand wordt centraalbandvakken.end()->geefvaknummer() aangeroepen maar dat kan toch niet omdat end() een iterator teruggeeft een voorbij het einde!
- In Centraalband::setModus zou gebruik gemaakt kunnen worden van for_each.
- Idem voor Sorteerder::Reset.
- De class Doos is wel erg leeg en zou dus door een simpele integer vervangen kunnen worden.
- De Random class maakt gebruik van de elementaire srand en rand functies. Het zou beter zijn (ook in verband met de uitbreidbaarheid en aanpasbaarheid) zou het beter zijn om de standaard C++ classes voor het genereren van random getallen te gebruiken.
- In Doosmaker::Wait wordt een new Doos aangemaakt. Maar deze door wordt nooit meer gedelete!
- Er worden ook andere objecten met new aangemaakt maar in het hele programma komt geen delete voor?