

# OO ontwerpen

- [2009 januari examen](#)
- [2011 januari examen](#)
- [2012 januari examen](#)
- [2013 januari examen](#)
- [2015 januari examen](#)
- [2016 januari examen](#)
- [2017 januari examen](#)
- [2018 januari examen](#)
- [2020 januari examen](#)
- [2020 juni examen](#)
- [algemeen](#)
- [2022 januari examen](#)

# 2009 januari examen

De vragen zijn redelijk uitgebreid op het examen, met wat code bij of UMLschema's daarom slechts de onderwerpen waarover de vragen gingen:

1. Decorator pattern
2. Template pattern
3. Singleton pattern
4. Naamloze implementatie van interface
5. Oefeningen op patterns
6. Collecties + oefeningen

# 2011 januari examen

1. Bespreek observer java built in en vergelijk met eigen implementatie. Geef voor en nadelen. (/10)
2. Een opgave waar je je eigen UML moet maken, patterns meegeven en uitleggen waarom. (/20)
3. Collections. De 4 main dingen Queue, list, set, map (/10)
4. Decorator typisch uml en per klasse uitleg geven (/10)
5. Pseudo code verwerken met solid (/10)
6. Java code i.v.m. singleton synchronized (/10)

# 2012 januari examen

## Vraag 1:

Gegeven: Situatieschets van een bedrijf waarvoor je een encryptiesoftware moet bedenken, die op drie verschillende manieren kan encrypteren/decrypteren.

- Ontwerp een klassendiagram (*eigen implementatie*)
- Som op welke principes/patterns je gebruikt hebt, en verduidelijk ze (*Waar in de code? Hoe toegepast?*)

## Vraag 2:

Gegeven: De klasse FitnessAbonnement van een plaatselijke fitness. Ze voorzien 3 extra's: groepslessen, DVDverhuur en frisdrank tijdens de fitness. Er is dan ook een methode voorzien om de prijs (met eventuele extra's) van het abonnement te berekenen.

- Geef aan welk design principle hier het meest geschonden wordt
- Welk design pattern kan dit probleem oplossen?
- Ontwerp een UML op basis van dat patroon
- Schrijf een paar regels code om jouw implementatie van de methode berekenPrijs() te verduidelijken

## Vraag 3:

Gegeven: een klassendiagram met een verduidelijkende tekst over de werking ervan.

- Som de aanpassingen ivm principes/pattern op die jij zou maken
- Voeg een gegeven externe klasse toe in het UML. Deze klasse komt van de opdrachtgever, en heeft dezelfde functionaliteit als jouw klassen, maar een andere naamgeving voor de methodes
- Moet de code worden aangepast om de externe klasse te implementeren? Verklaar.

# 2013 januari examen

Er waren drie grote vragen en enkele kleinere vragen bij.

- De eerste vraag moest je een klassediagram verbeteren, de principes erop toepassen en patronen.
- De tweede vraag was rondom code en welke fouten erin stonden. Best kon je hiervoor een klassediagram maken om dan te zien wat er fout loopt.
- De derde vraag kreeg je een situatie en moest je gewoon het model gedeelte uitwerken in een klassediagram.

# 2015 januari examen

Drie grote vragen.

## Vraag 1:

- MVC wordt een compound pattern genoemd (samengesteld patroon). Welke patronen bezit MVC? (niet gewoon het acroniem voluit schrijven)
- Gegeven een onoverzichtelijk klassendiagram, duid voor elke klasse aan of deze bij model, view of controller hoort.
- Een gebruiker drukt op een knop. Duid in dit klassendiagram aan elke stap die vervolgens wordt uitgevoerd en door welke klasse.
- Schrijf deze stappen stap voor stap op.

## Vraag 2:

Een gebruiker vraagt een programma waarbij hij een even aantal getallen wilt ingeven en pas stopt bij een willekeurig getal. Deze lijst kan dan eventueel gemanipuleerd worden (sorteren, schudden, ...). Deze eventuele gemanipuleerde lijst kan dan wiskundige bewerkingen ondergaan bij het uitprinten (per paar van 2 optellen of aftrekken of vermenigvuldigen, ...). Een inleescode voor de lijst getallen is gegeven.

De vraag luidt: schrijf door middel van een template een implementatie van dit programma. Programma 1 moet dan gewoon inlezen en elk paar van 2 optellen. Programma 2 moet ze ook nog eens sorteren en langs achter beginnen om elk paar af te trekken. (voorbeeld output programma 2: rij getallen = 2 4 5 6 8 1 0 3 5 4, sorteren = 0 1 2 3 4 4 5 5 6 8, output = 2 0 0 1 1)

## Vraag 3:

Gegeven is een uitleg over een schaakprogramma met een beknopt klassendiagram. Men wilt het schaakprogramma uitbreiden, hiervoor wordt een tekst voorzien. Geef voor elk patroon in de tekst een UML en de motivatie.

# 2016 januari examen

Drie grote vragen.

## **Vraag 1:** (/5)

- Gegeven een vereenvoudigd UML van het spel 2048, uitleg over hoe de applicatie uitgebreid moet worden.
- Gevraagd: Teken UML en leg uit over hoe volgende patronen hier specifiek toegepast kunnen worden en wat hun doel is.

Observer, factory, state

## **Vraag 2:** (/3)

- Gegeven een uitleg over inlezen van tekstbestand, filteren, sorteren en uitprinten.

Ook basisimplementatie van voorgaande gegeven.

- Gevraagd: Schrijf de code van het Template en Strategy patroon dat hiervoor toegepast moet worden, dus strategy in template.

## **Vraag 3:** (/3)

- Gegeven vereenvoudigd UML over wiskundeoefeningen.
- Gevraagd: Duid op het UML aan welke delen tot model, view en controller behoren. Toevoegen van een functie, pas de klassen die aangepast moeten worden aan.

# 2017 januari examen

## Vraag 1:

Gegeven een uitleg en een uitgebreid UML van een winkelapplicatie

- Geef twee patronen die in het compound patroon MVC gebruikt worden
- Duid deze aan op het UML
- Geef voor elke klasse van het UML aan of het Model, View or Controller deel is
- Stel een gebruiker besteld een product geef een lijst van de opeenvolgende klasse en methodes die worden gebruikt om dit uit te voeren (en geef nummers aan in het UML)

## Vraag 2:

Gegeven een uitleg, state chart diagram, een test en output van een poort in een metrostation. Geef de code voor de toegangspoort en de states om de gegeven output van de test te krijgen. (Dit is ongeveer de opgave, maar herinner mij niet meer 100%) Statechart:

[OOO\\_2017\\_State\\_toegangspoort.jpg](#)

Test:

```
ToegangsPoort poort = new ToegangsPoort();
poort.scan(new ToegangsBewijs(7));
poort.doorlopen();
//Geldig toegangsbewijs scannen en doorlopen
poort.scan(new ToegangsBewijs(0));
poort.doorlopen();
//Geen beurten meer
poort.doorlopen();
//doorlopen zonder te scannen
ToegangsBewijs bewijs = new ToegangsBewijs(4);
poort.scan(bewijs);
poort.scan(bewijs);
poort.doorlopen();
//2 keer achter elkaar scannen
poort.afsluiten();
poort.scan();
```

Output:

```
"Geldig bewijs - Loop door"
"Reiziger loopt door"
```



"Geen beurten meer"

"ALARM!"

"ALARM!"

"Geldig bewijs - Loop door"

"Terugbetaling gebeurt"

"Reiziger loopt door" (Denk dat dit een fout is, heb ook zo op het examen gezegd anders kun je gewoon 2 keer scannen en gratis doorlopen)

"Einde van de dag - aantal Reizigers: 4" (Is hoe vaak er gescand is, maar eigenlijke aantal reizigers is 2, ook een fout denk ik...?)

"Metrostation is dicht"

### Vraag 3:

Gegeven een sort methode:

```
public Person [] sort(Person[] persons, String sortalgoritme, Comparator<Persons> comparator) {  
    Person[] sortedpersons = persons;  
    if (sortalgoritme.equals("Quick") {  
        //sort using quicksort algoritme  
    }  
    if (sortalgoritme.equals("Random") {  
        //sort using randomsort algoritme  
    }  
    return sortedPersons;  
}
```

- Welk principe wordt hier het meest geschonden?
- Hoe zou je het beter doen en teken UML class diagram.

# 2018 januari examen

## Vraag 1:

Gegeven code van een klasse om een bestand in te lezen per letter (character), code van een klassen om tekst (String) letter per letter uit te lezen, code van de launcher, een UML-diagram van het decorator-patroon. Men wilt bovenop het inlezen een bestand/tekst, deze kunnen manipuleren, zo bijvoorbeeld de tekst omzetten naar camelcase, spaties tussen iedere letter plaatsen, alle spaties verwijderen...

- Ontwerp met behulp van het decorator-patroon een UML om dit te implementeren.
- Schrijf de code van de nieuwe klassen uit
- Schrijf de code van de launcher uit als je in de tekst eerst alle spaties wilt verwijderen, dan in camelcase wilt zetten en tenslotte tussen iedere letter een spatie wilt plaatsen (dit is een zin -> ditiseenzin -> Ditiseenzin -> D i t i s e e n z i n)

## Vraag 2:

Gegeven een UML-diagram voor het spel Tic-Tac-Toe, dat het MVC model volgt.

- Welke patronen herken je in dit model? Benoem ze en zeg welke klassen er deel van uit maken
- Duid voor iedere klasse aan of het Model, View of Controller is
- Stel een user klikt een vak aan; Welke methodes worden er achter elkaar uitgevoerd, en welke klassen roepen deze methodes aan?

## Vraag 3:

Gegeven een stuk code, ruwweg de volgende:

```
public class Compression {  
    public void compress(List<File> files, File targetFile, String algorithm) {  
        if (algorithm.equals("zip")) {  
            // Do ZIP compression  
        } else if (algorithm.equals("rar")) {  
            // Do RAR compression  
        } else {  
            throw new IllegalArgumentException("Algorithm not found");  
        }  
    }  
}
```

- Welk principe wordt hier het meest geschonden en waarom?
- Hoe zou je dit oplossen? Teken een UML-diagram.

# 2020 januari examen

Examen bestond uit 4 vragen en je had 2h30 tijd.

## Vraag 1: (/2)

- Gegeven: een stuk code

```
public Player {  
    playCD()  
    showMovie()  
    playRadio(int frequentie)  
}  
  
public DVDPlayer implements Player {  
    public playCD() { //implementation }  
    public showMovie() { //implementation }  
    public playRadio(int freq) { throw error }  
}
```

- Welk principe wordt hier het meest geschonden en waarom?
- Maak een UML die het beter zou doen (principe toepassen)

## Vraag 2: (/3)

- Gegeven: Tic-Tac-Toe verhaal
- Maak UML met Observer & MVC
- Geef een lijst met alle functies die worden uitgevoerd wanneer er op "increment" wordt gedrukt (in het eerste view)
- Schrijf code van de eerste controller (die van het eerste view)

## Vraag 3: (/5)

- Gegeven: verhaal van een ATM en een statechart
- Maak UML met als pattern State
- Schrijf code van alle klassen, interfaces, ...

## Vraag 4: (/5)

- Gegeven: verhaal van een Fitnessabonnement

```
public Abonnement {  
    constructor  
    getters en setters van frisdranken handdoeken en groepsSport  
    public getPrijs() {  
        int prijs = 30;  
        if (groepsSport)  
            prijs+=10;  
        if (frisdranken)  
            prijs+=5;  
        if (handdoeken)  
            prijs += 15;  
    }  
}
```

- Maak UML met als pattern Decorator + uitbereiding korting met Strategy
- Schrijf code voor de launcher (voor die specifieke situatie)

# 2020 juni examen

## **theorie:**

letterlijk theorie vragen van dat word document

-wat zijn de 3 wrapper classes en waar verschillen zij in

-vergeten

## **praktijk:**

1 grote oefeningen waar je 3-4 patronen in UML moest tekenen

maak gebruik van !MVC model, strategy,observer,factory,..?

verhaaltje over state en hier UML van maken en paar klassen uitschrijven(punten pakker)

verhaalte over decorator toepassen, uml tekenen en paar klassen uitschrijven (het ging over ijsjes, sommige willen ijsje met saus, nootjes en andere ijsje met snoep, nootjes,snoep (punten pakker)

groetjes aan de familie, kirito-kun

# algemeen

## **Vanaf 2019-2020**

Dit vak wordt gegeven door Patrick Fox, die zijn vak erg goed beheerst. Er zijn twee sessies in de week: eenmaal een praktische zitting om aan de projecten te werken en eentje waarbij wat concepten uit het handboek verder besproken worden omdat die nodig zijn voor het succesvol uitwerken van het project.

Tegen het einde van het semester wordt er verwacht dat je in groep een programmeeropdracht uitvoert. Het project begint na de herfstvakantie en je krijgt enkel user stories en er wordt bij elke user story aangegeven waar/hoe je patronen moet toepassen. Hiervoor moet je ook een vrij uitgebreid verslag maken (met mooie UML). Dit project moet tegen de Kerstvakantie af zijn en staat op 25% van de punten. Tijdens het examen is er ook een theoriegedeelte dat ook meetelt voor 25%. Het praktijkgedeelte van het examen staat dus nog op 50%, volledig schriftelijk en gaat over het toepassen van de patronen (zie vragen laatste paar jaren).

## **Tot 2018-2019**

Dit vak wordt o.a. gegeven door Patrick Fox, die zijn vak erg goed beheerst. Er zijn twee sessies in de week: eenmaal een praktische zitting om aan de projecten te werken en eentje waarbij wat concepten uit het handboek verder besproken worden omdat die nodig zijn voor het succesvol uitwerken van de projecten.

Doorheen het semester krijg je twee programmeeropdrachten die je in groep moet uit te voeren. Eén voor de herfstvakantie waarbij er steeds gezegd wordt waar/hoe je het patroon van die week moest toepassen. Dit project wordt op het einde nagekeken door Fox, maar staat niet op punten. Het tweede project begint na de herfstvakantie en je krijgt enkel use stories en je moet zelf zien waar/hoe je patronen wilt toepassen. Hiervoor moet je ook een vrij uitgebreid verslag maken (met mooie UML). Dit project moet tegen de Kerstvakantie af zijn en staat op 25% van de punten. Halverwege het semester is er ook een theorietest die ook meetelt voor 25%. Het examen in Januari staat dus nog op 50%, volledig schriftelijk, 3u de tijd en gaat over het toepassen van de patronen (zie vragen laatste paar jaren).

# 2022 januari examen

## Theorie:

letterlijk theorie vragen van dat word document

- 1 Gaten kazen tekst waar je de 2 principles en 2 patterns moest in invullen aan de hand van de rest van de zin
- 2 Een UML state diagram met wat uitleg tekst die je dan moest omzetten naar een UML class diagram, hierin zat een abstracte classe
- 2.1 De abstracte classe hun java code schrijven

## Praktijk:

- 1 UML classe diagram maken van een groot probleem omschrijving
- 2 Wanneer er op een knop in de probleem omschrijving wordt gedrukt, wat zal er dan gebeuren, schrijf lijn per lijn de weg op dat het afgelegd (dus naar welk bestand het gaat)
- 3 De Launchclasse van de probleem omschrijving opschrijven
- 4 Java code van een controller opschrijven

-Succes xoxo