Examenvragen Advanced Programming Topics / Software methodologies

Opgesteld door Seppe Snoeck en daarbij dus geen officiële examen vragen!

1. **Wat is het Collection framework?**
2. **Op welke manier ken je een Collection doorlopen?**
3. **Wat zijn de voordelen van Generics?**
4. **Geef 5 naming conventions.**
5. **Wat zijn generics?**
6. **Wat stelt een wildcard voor? + symbool?**
7. **In welke situaties kan een wildcard gebruikt worden?**
8. **Wat is een Upper Bounded Wildcard?**
9. **Wat is een Lower Bounded Wildcard?**
10. **Leg kort uit. Set, SortedSet, Queue, Map en SortedMap.**
11. **Hoe itereer je over een Map?**
12. **Wat is het snelst en waarom? Een TreeSet of een ArrayList?**
13. **Wat is het doel van sorteren van data?**
14. **Wat beïnvloed de snelheid van sorteren?**
15. **Wat is het slechtste sorteer algoritme en leg hem uit.**
16. **Leg insertion en bubble sort uit.**
17. **Leg merge en quick sort uit.**
    Waarom zou je aan Test Driven Development willen doen?
18. **Wat is de ROI van TDD?**
19. **Waarom zou je testen willen automatiseren?**
20. **Wat zijn de voordelen van TDD?**
21. **Welke strategie neem je aan om testen te schrijven?**
22. **Op welke zaken kan je zoal testen?**
23. **Leg TDD uit aan de hand van red/green bar.**
24. **Wat is het verschil tussen black en white box testing?**
25. **Hoe behaal je een ISO9001 certificaat en hoe behoud je deze?**
26. **Wat concludeer je van ISO9001?**
27. **Bespreek het ISO9126-1 en waarvoor staat het?**
28. **Wat is Agile vooral niet maar wat wel?**
29. **Wat zijn de 12 principes van Agile?**
30. **Wat is het verschil tussen waterfall en scrum?**
31. **Wat zijn de scrum rollen?**
32. **Wat zijn de artefacten van scrum en leg hun functie uit.**
33. **Welke Scrum ceremonies bestaan er?**
34. **Hoe verloopt Scrum?**
35. **Waarvoor staat BPM?**
36. **Hoe komen bedrijven aan de top binnen hun niche?**
37. **Wat is business process?**
38. **Hoe kan jij je business verbeteren?**
39. **Wanneer weet je niet wat je aan het doen bent?**
40. **Wat is het moeilijkste: Op ideeën komen of 'get shit done'?**
41. **Hoe effectief is een organisatie?**
42. **Wat is Contiuous Integration?**
43. **Geeft enkele typische CI challenges.**

1. Wat is het Collection framework?
Het Collection framework omvat datastructuren die meerde elementen groepeert tot één geheel.

2. Op welke manier ken je een Collection doorlopen?
Je kan een Collection doorlopen op 3 manieren: via een foreach loop, interaction of aggregation functions.

3. Wat zijn de voordelen van Generics?
Een betere type control voor de compiler. Het is immers “gemakkelijker” om compile fouten op te lossen dan runtime fouten.
Ook is er geen casting meer nodig.

4. Geef 5 naming conventions.
Element, Type, Key, Value & Number

5. Wat zijn generics?
Generics laat toe om types als parameters te gebruiken bij het definiëren van klassen, interfaces & methoden. Hiermee kan heel veel dubbele code vermeden worden.

6. Wat stelt een wildcard voor? + symbool?
Een wildcard stelt een onbekend type voor met het symbool “?”.

7. In welke situaties kan een wildcard gebruikt worden?
Een wildcard kan in 3 situaties gebruikt worden: Als type van een parameter, veld en als lokale variable.

8. Wat is een Upper Bounded Wildcard?
UBW’s worden gebruikt om de restricties van een variable te verzachten.
List< ? extends Number>

9. **Wat is een Lower Bounded Wildcard?**
LBW’s beperken zicht tot het onbekende type, specifieke type of tot een super type van dat type.
List<? super A>

10. **Leg kort uit. Set, SortedSet, Queue, Map en SortedMap.**
Set is een Collection die geen duplicaten bevat.
SortedSet is een set die zijn elementen in oplopende volgorde bewaard.
Queue doet aan FIFO (First In, First Out.)
Map belt zijn sluitels af op een waarde.
SortedMap is een map die zijn elementen bewaard in oplopende volgorde.

11. Hoe itereer je over een Map?
```
For(Keytype key : map.keyset()) {
        System.out.println(key);
}
```

12. Wat is het snelst en waarom? Een TreeSet of een ArrayList?
Een TreeSet is het snelst omdat hij zijn elementen gesorteerd bijhoud.

13. Wat is het doel van sorteren van data?
Het doel is om zo snel mogelijk te sorteren en om daarvoor zo min mogelijk geheugen te gebruiken.

14. Wat beïnvloed de snelheid van sorteren?
Het aantal loops en het aantal elementen die doorlopen moeten worden.

15. Wat is het slechtste sorteer algoritme en leg hem uit.
Bogosort is het slechtste sorteer algoritme en hij werkt als volgt.
Stel je hebt een boek speelkaarten. Eerst ga je na of het boek gesorteerd is. Is dit gesorteerd -> Dan heb je chance. Indien het niet gesorteerd is dan hooi je het boek in de lucht en raap je alle speelkaarten in willekeurige volgorde van de groep en stel je een “nieuw” boek samen. Dan herhaal je alle stappen. Is het boek gesorteerd?

16. Leg insertion en bubble sort uit.
Insertion sort: We nemen aan het het eerste element altijd gesorteerd is. Daarna bekijken we het volgende ongesorteerde element en vergelijken we die met get gesorteerde deel van de elementen en **verschuiven** we de elementen om het niet gesorteerde element in het gesorteerde deel te plaatsen.
Bubble sort: Bij bubble sort werk je met een “swap counter”. De elementen zijn niet gesorteerd zolang deze niet op 0 staat na het doorlopen van alle elementen. Aan de start van het sporten zet je de router op 0. Dan **vergelijk je 2 elementen per keer** met elkaar en schik je hen volgens de vereisten. Zo zal door de eerste iteratie het grootste element achteraan staan, als men sorteert van klein naar groot. Als 2 elementen van positie moeten veranderen dan voegen we 1 to aan de swap counter. Dit wordt herhaald tot de swap counter op 0 staat en dan zijn we zeker dat de elementen gesorteerd zijn.

17. Leg merge en quick sort uit.
Beiden zijn recursieve sorteer algoritmen.
Merge sort: De collectie/array van elementen wordt **opgesplitst tot individuele elementen**. Vergelijk 2 elementen met elkaar, sorteer en **merge** hen in tijdelijke arrays.Nu doe je het zelfde met de nieuwe rijen. Je vergelijkt hen en sorteert hen. Dit process herhaal je tot er 1 array overblijft. Dan heb je een gesorteerde collectie/array.
Quick sort: Quick sort maakt gebruik van een **pivot** wat een element is van de array en die aan volgende condities moet voldoen nadat we gesorteerd hebben:

1. De pivot is in zijn correcte positie in een gesorteerde array.
2. De elementen links zijn kleiner.
3. De elementen rechts zijn groter.

Eerst kies je een pivot en plaats je hem aan het einde van de array. Vervolgens zoeken we 2 elementen. Startende van links nemen we het eerste element dat groter is dan de pivot en startende van links de eerste die kleiner is dan de pivot. en verwissel deze van plaats. Herhaal dit tot dat het rechtse element groter is dan het linkse. Dan verwissel je het linkse element met de pivot. En nu is onze pivot op zijn juiste positie in de array. Nu sorteer je het linkse deel en het rechtse deel van de pivot op de zelfde manier tot alles gesorteerd is.

Hoe kies je het beste een pivot?
Median-of-three is een veel gebruikte methode. Je selecteert het 1e, middelste en laatste element in de array en sorteert deze door van plaats te verwisselen en dan het middelste element is onze pivot. deze heeft veel kans om de mediaan te zijn om zo het werk evenredig te verdelen. Een pivot kiezen kan veel effect hebben op de uitvoering van het algoritme.

18. Waarom zou je aan Test Driven Development willen doen?
Als je kijkt naar het waterfall model dan komt testen pas op de laatste plaats net als in de meeste gevallen. Door een slechte tijd/budget ratio is de periode van testen verkleind of er word zelfs niet getest.
Start vroeg met testen om een “time squeeze” te verkopen, sneller defecten te vinden, op de juiste moment te testen en om vertrouwen op te bouwen.

19. Wat is de ROI van TDD?
- Tragere ontwikkeling
- Hogere kwaliteit van code
- Verkort de leveringstijd ven het “afgewerkt” product

20. Waarom zou je testen willen automatiseren?
Omdat testen saai, nodig en vooral repetitief is. Er moet bij elke wijziging getest worden en manueel testen omvat enkel specifieke waarden die huidige fouten veroorzaken en is enorm traag. Met automatische testen kan je er zoveel runnen als nodig, wanneer het nodig is, altijd op dezelfde manier, met zoveel mogelijk verschillende waarden en kan bovendien door een bot gedaan worden.

21. Wat zijn de voordelen van TDD?
Testen zijn herhaalbaar, deels documentatie, support, nooit meer dezelfde bug en je spendeert minder tijd in deug mode.

22. Welke strategie neem je aan om testen te schrijven?
Je kiest een postcondition en maakt van een specificatie een test. kan je geen test maken voor een specificatie dan was deze niet duidelijk genoeg.

23. Op welke zaken kan je zoal testen?
Op validatie om aan te tomen dat de code voldoet aan de voorwaarden, op defecten en exceptions.

24. Leg TDD uit aan de hand van red/green bar.
1. Schrijf test
2. Schrijf minimaal skelet om de test te compileren
   Redbar
3. Schrijf code om de test te runnen
   Greenbar
4. Refactor
5. Test code
   Greenbar
6. Herhaal

25. Wat is het verschil tussen black en white box testing?
Bij blackbox testing heb je geen kennis over de interne structuur en is geschreven naar de specificaties toe met focus op de eindgebruiker. Dit leid tot 80% coverage.
White box testing is geschreven naar de implementatie met focus op de ontwikkelaars. Dit omvat de resterende 20% code die mist bij black testing.

26. Hoe behaal je een ISO9001 certificaat en hoe behoud je deze?
Eerst en vooral voorzie je documentatie met opvolgend een pre audit. Ja krijgt een certification audit die dan voor 3 jaar geldig is. in die 3 jaar krijg he nog 2 opvolgende audits (elk jaar). Na 3 jaar vervalt je certificaat en dan wordt het process herhaald vanaf de certification audit. Kort: elk jaar mag je een audit verwachten.

27.Wat concludeer je van ISO9001?
De ISO9001 is een “quality management framework” die als handleiding kan gebruikt worden. Het is vooral gericht naar de klas toe. Het is flexibel en generiek.

28. Bespreek het ISO9126-1 en waarvoor staat het?
De IOS9126-1 staat voor “Software quality model” en omvat volgende zaken:
Functionality, reliability, usability, efficiency, maintainability en portability.
(De inhoud moet je zelf maar bekijken)

29. Wat is Agile vooral niet maar wat wel?
Agile is geen methodology, specifieke manier om software te ontwikkelen, framework of process. Agile is een set van waarden en principes.

30. Wat zijn de 12 principes van Agile?
Deze principes geven jou de mogelijkheid om goede keuzes te maken.
1. Onze hoogste prioriteit is om de klant tevreden te stellen doorheen het jaar en om continu waardevolle software te leveren.
2. Verwelkom veranderende vereisten, zelfs laat in de ontwikkeling.
3. Lever frequent werkende software in een tijdspanne an een aantal weken tot een aantal maanden maar korte periodes krijgen de voorkeur.
4. Ondernemers en ontwikkelaars moeten dagelijks met elkaar samen werken doorheen het project.
5. Maak projecten rondom gemotiveerde individuen. Geef hen de omgeving en ondersteuning die ze nodig hebben om de klus te klaren.
6. De meest efficiënte manier is face-to-face communicatie
7. Werkende software is de primaire meetbaarheid van processie.
8. Agile promoot duurzame ontwikkeling. De sponsors, ontwikkelaars en gebruikers zouden een constante vooruitgang moeten boeken voor onbepaalde duur.
9. Continue aandacht voor technische excellentie en een goed ontwerp verbetert de agility(wendbaarheid).
10. Simpelheid - De kunst in het maximaliseren van het werk dat niet klaar is, is essentieel.
11. De beste architecturen, requirements en designs komen voort uit zelf organiserende teams.
12. Op regelmatige tijdsstippen evalueert het team zich zelf hoe ze effectiever kunnen werken.

! Het echte doel van Agile is om mensen een fundering te geven om beslissingen te maken over hoe ze best software ontwikkelen.

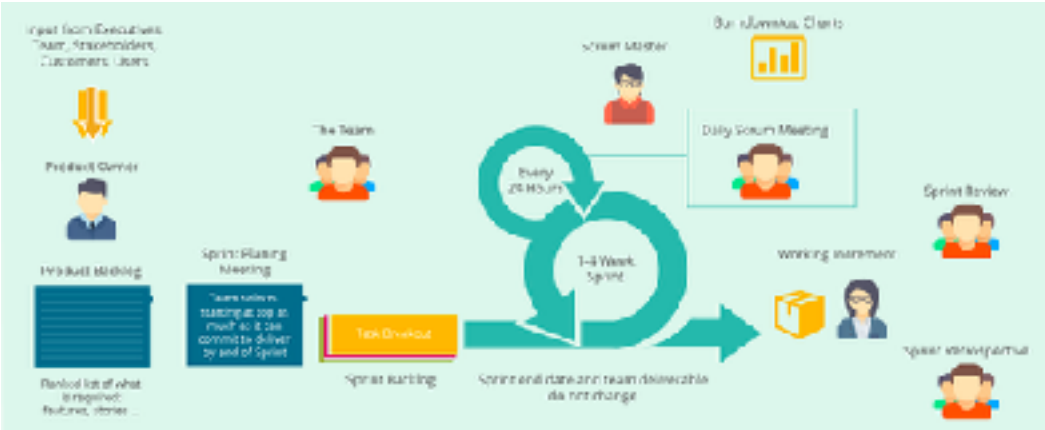
31. Wat is het verschil tussen waterfall en scrum?
Binnen waterfall wordt er in stages gewerkt. Eerst wordt er gepland, dan ontwikkeld, nadien worden testen gedaan, het product wordt beoordeeld en daarna gelanceerd. Als er dan een fout of ondeugendheid wordt opgemerkt wordt die verschoven naar het vorig stadium enz. Zo kan je bv in een oneindige loop tussen ontwikkelen en testen zetten zonder te product te lanceren.
Scrum neemt een andere aanpak. Binnen scrum wordt er gestraft naar kleine releases. Per release wordt er gepland, ontwikkeld, getest en beoordeeld voor specifieke zaken. Zo een release heet men een sprint. Zo worden kleine release een totaal “afgewerkt” product. een sprint loop meestal over 2 weken, dit kan variëren.

32. Wat zijn de scrum rollen?
Product owner, Scrum master & het dev team

33. Wat zijn de artefacten van scrum en leg hun functie uit.
Product backlog: De PO maakt een lijst met features en hun prioriteit. Een feature word beschreven als een user story.
Sprint backlog: Dit is een lijst de samen gesteld wordt in retrospective. Men neemt zaken uit de product backlog om te ontwikkelen in de eerst komende sprint.
Burndown chart: Toont processie aan tijdens de sprint over bepaalde taken. Deze zou leeg moeten zijn wanneer het product volledig klaar is. Het is een visuele representatie van werk te doen over een tijdsspanne.

34. Welke Scrum ceremonies bestaan er?
Sprintplanning: PO, SM, Team -> de user stories worden besproken en hun gewicht.
DailyScrum: Wat is er klaar sinds de vorige meeting? Waar werk je aan? Waar heb je hulp nodig?
Sprint review & retrospective: Einde van de sprint, demo aan PO, Wat doen we volgende sprint?

35. Hoe verloopt Scrum?
Product backlog -> Sprint planning -> Sprint backlog ->Sprint (tijdens de sprint Daily scrum) -> potentieel shippable product ->Sprint review en retrospective



36. Waarvoor staat BPM?
Business Process Management

37. Hoe komen bedrijven aan de top binnen hun niche?
Door hun focus te verkleinen. Bv. Enkel mountainbikes ontwikkelen ipv alles soorten.

38. Wat is business process?
Een groep gerelateerde activiteiten die samen een waarde vormen voor de klant.

39. Hoe kan jij je business verbeteren?
Map, analyse, redesign, aquire, implement & communicate, review

40. Wanneer weet je niet wat je aan het doen bent?
Als je wat je aan het doen bent niet kan beschrijven als een proces dan weet je niet wat je aan het doen bent.

41. Wat is het moeilijkste: Op ideeën komen of 'get shit done'?
Get shit done

42. Hoe effectief is een organisatie?
Een organisatie is maar zo effectief als zijn processen.

43. Wat is Contiuous Integration?
Het is een software ontwikkelen proces waar ontwikkelaars meermaals hun verandering aan code morgen in de centrale repo. Die nadien automatisch build en zijn testen doorloopt.

44. Geeft enkele typische CI challenges.
- Frequent committen
- Onderhouden van een 'single source code'
- Automatiseren van testen en builds
- Testen in een gelijkaardige omgeving als die van de productie
- Het proces visueel toonbaar maken aan het team