

## **Eksamensspørgsmål i I4SWD**

### **1. SOLID 1: SRP, ISP og DIP**

- Redegør for designprincipperne Single Responsibility Principle (SRP), Interface Segregation Principle (ISP) og Dependency Inversion Principle (DIP)
- Redegør for, hvordan du mener anvendelsen af principperne fremmer godt software design
- Vis et eksempel på anvendelsen af et eller flere af principperne i software design.
- Redegør for konsekvenserne ved anvendelsen af SRP, ISP og/eller DIP – har det nogle ulemper?

### **2. SOLID 2: OCP, LSP og DIP**

- Redegør for Open-Closed Principle, Liskov's Substitution Principle (ISP) og Dependency Inversion Principle (DIP)
- Redegør for, hvordan du mener anvendelsen af principperne fremmer godt software design
- Vis et eksempel på anvendelsen af et eller flere af principperne i software design.
- Redegør for konsekvenserne ved anvendelsen af OCP, LSP og/eller DIP – har det nogle ulemper?

### **3. Patterns 1: GoF Strategy + GoF Template Method**

- Redegør for, hvad et software design pattern er.
- Sammenlign de to design patterns GoF Strategy og GoF Template Method – hvornår vil du anvende hvilket, og hvorfor?
- Vis et designeksempel på anvendelsen af GoF Strategy
- Redegør for, hvordan anvendelsen af GoF Template fremmer godt software design.
- Redegør for, hvilke(t) SOLID-princip(per) du mener anvendelsen af GoF Strategy understøtter

### **4. Patterns 2: GoF Observer**

- Redegør for, hvad et software design pattern er.
- Redegør for opbygningen af GoF Observer.
- Sammenlign de forskellige varianter af GoF Observer – hvilken vil du anvende hvornår?
- Redegør for, hvordan anvendelsen af GoF Observer fremmer godt software design
- Redegør for fordele og ulemper ved anvendelsen af GoF Observer
- Redegør for, hvilke(t) SOLID-princip(per) du mener anvendelsen af GoF Observer understøtter

### **5. Patterns 3: GoF Factory Method/Abstract Factory**

- Redegør for, hvad et software design pattern er.
- Redegør for opbygningen af GoF Factory Method og GoF Abstract Factory.
- Giv et designeksempel på anvendelsen af GoF Abstract Factory.

## **6. Patterns 4: State patterns**

- Redegør for, hvad et software design pattern er.
- Redegør for strukturen i GoF State Pattern
- Sammenlign switch/case-implementering med GoF State
- Redegør for fordele og ulemper ved anvendelsen af GoF State.
- Redegør for, hvordan et UML (SysML) state machine diagram mapper til GoF State.

## **7. Patterns 5: Model-View-Controller og Model-View-ViewModel**

- Redegør for, hvad et software design pattern er.
- Redegør for Model-View-Control mønstret og dets variationer.
- Redegør for Model-View-Presenter mønstret og dets variationer.

## **8. Patterns 6: Model-View-ViewModel**

- Redegør for, hvad et software design pattern er.
- Redegør for Model-View-ViewModel mønstret og dets variationer.

## **9. Patterns 7: Redegør for følgende concurrency mønstre**

- Parallel Loops
- Parallel Task

## **10. Patterns 8: Redegør for følgende concurrency mønstre**

- Parallel Aggregation
- MapReduce

## **11. Patterns 9: Redegør for følgende concurrency mønstre**

- Futures
- Pipelines

## **12. Software arkitektur**

- Redegøre for begrebet softwarearkitektur.
- Hvordan er den typiske softwarearkitektur?
- Hvordan udarbejdes en software arkitektur?
- Hvorledes dokumenteres en software arkitektur?
- Hvorledes udarbejdes og dokumenteres en concurrency model?

## **13. Obligatorisk opgave**

- Redegør for indholdet af din egen obligatoriske opgave