

Tentamensövning

Objekt-orienterad programmering och design

Alex Gerdes, 2016

Tentamen

- När: Tisdag, 10/1, kl 14:00-18:00
- Var: SB
- Hjälpmedel: inga
- Bifogas:
 - Alla nödvändiga API som behövs för uppgifterna.
 - UML syntax "cheat sheet"

Upplägg

- Tekniska aspekter:

- Polymorfism:
 - Interfaces
 - Generics
 - Overloading
- Arv:
 - Abstract classes
 - Overriding, dynamisk bindning
- Variance
- Exceptions
- Objects vs Classes
- Trådar

- Design-aspekter:

- Principer
 - SRP, OCP, LSP, ISP, DIP, CQSP
- Design patterns
 - MVC
 - Observer, Decorator
 - ...
- Tekniker
 - Immutability, defensive copying
 - Method cascading

Typfråga: teknisk

1. Betrakta deklarationerna av variabler. Ange sedan för var och en av följande statements om det är:

- Korrekt.
- Ger kompileringsfel.
- Ger runtime-fel (exception).

```
List<Object> listObject;  
List<Number> listNumber;  
List<? extends Number> listExtendsNumber;  
List<? super Number> listSuperNumber;  
List<Double> listDouble;  
List<? extends Double> listExtendsDouble;  
List<? super Double> listSuperDouble;  
Number n;  
Double d;  
Integer i;
```

```
a) listNumber.add(d);  
b) i = listExtendsNumber.get(0);  
c) listNumber = listDouble;  
d) ...
```

Obs: Många fler kombinationer på tentan.

Typfråga: design

1. Vilken princip strider koden mot?
2. Åtgärda detta genom att använda designmönstret *Strategy Pattern*.
 - a) Rita ett UML-diagram över din nya lösning.
 - b) Visa den nya implementationen av metoden play.
 - c) Visa implementationen av en strategi.

```
public class MusicPlayer {  
    Object o;  
    ...  
    public void play() {  
        if (o instanceof Trumpet) {  
            ((Trumpet) o).playTrumpet();  
        } else if (o instanceof Piano) {  
            ((Piano) o).playPiano();  
        } else if (o instanceof Guitar) {  
            ((Guitar) o).playGuitar();  
        }  
    }  
}
```

Obs: På tentan hade ni fått hela koden.

Rekommenderade uppgifter idag

- TDA550 Augusti 2015
 - Uppgift 1: UML, arv, static vs dynamic type
 - Uppgift 3: Apply design pattern
 - Uppgift 6: Apply design pattern
 - Uppgift 8: Use of generics, variance
 - Uppgift 9: Apply design pattern
 - Uppgift 10: Use of threads