

Objekt-orienterad Programmering och Design

TDA551

Alex Gerdes, HT-2016

Kursteamet

- Dr. Alex Gerdes – kursansvarig, föreläsare
- Dr. Niklas Broberg – examinator, (föreläsare)

- Fredrik Sjöholm – handledare
- Johan Andersson – handledare
- Sebastian Norlin – handledare
- Christoffer Medin – handledare
- Gustav Svensson – handledare

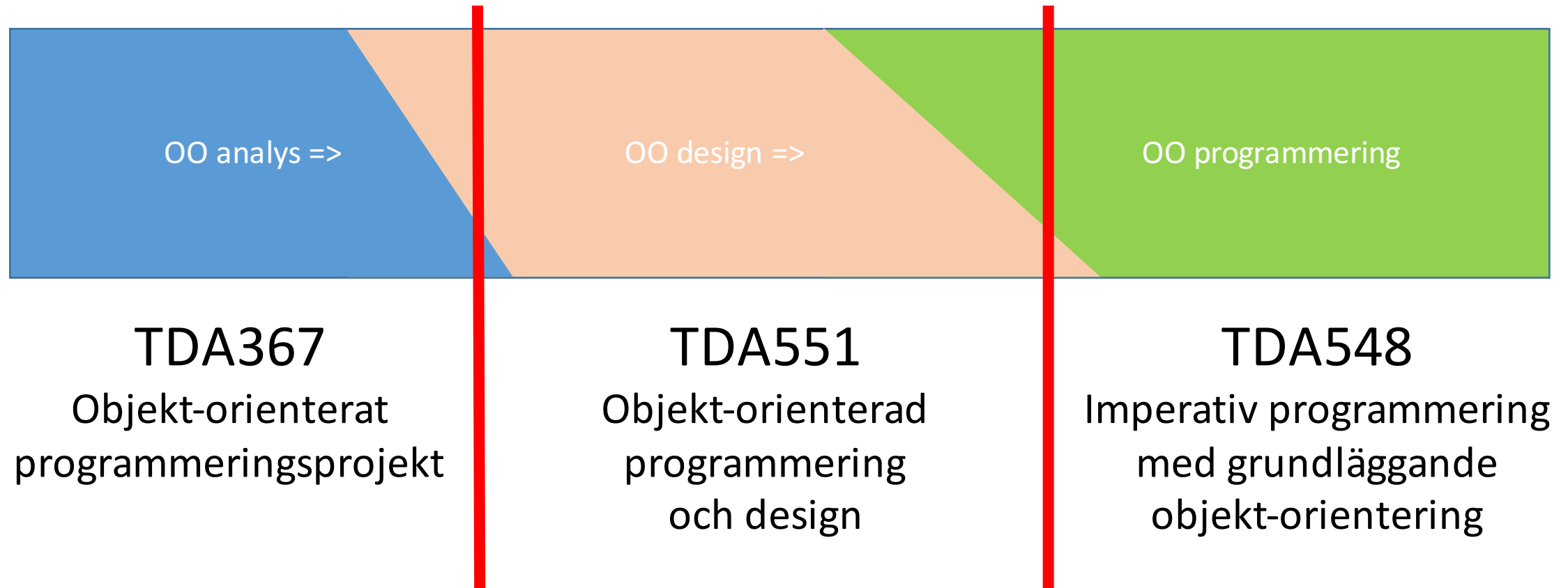
Vad är "objekt-orientering"?

- ... objekt-orienterat språk?
- ... objekt-orienterat program?
- ... objekt-orienterad design?

Objekt-orientering är...

- ett sätt att *modellera* en applikation i termer av kommunicerande objekt:
 - Objekt-orienterad *design* är skapandet av modellen
 - Objekt-orienterad *programmering* är implementationen av modellen
 - Ett "objekt-orienterat" *språk* är ett programspråk med inbyggt stöd för att underlätta implementationssteget
- Ett program skrivet i ett "objekt-orienterat" språk måste inte vara objekt-orienterat
- Ett objekt-orienterat program måste inte implementeras i ett "objekt-orienterat" språk

Objekt-orientering: process



Verktyg vs principer

- Ni har lärt er *hur* olika verktyg (i Java) fungerar
- I den här kursen vill vi svara på frågor som *när* ska olika verktyg bör användas, *varför* de bör användas, och hur de bör användas *bäst*

THE LIFE OF A SOFTWARE
ENGINEER.

CLEAN SLATE. SOLID
FOUNDATIONS. THIS TIME
I WILL BUILD THINGS THE
RIGHT WAY.

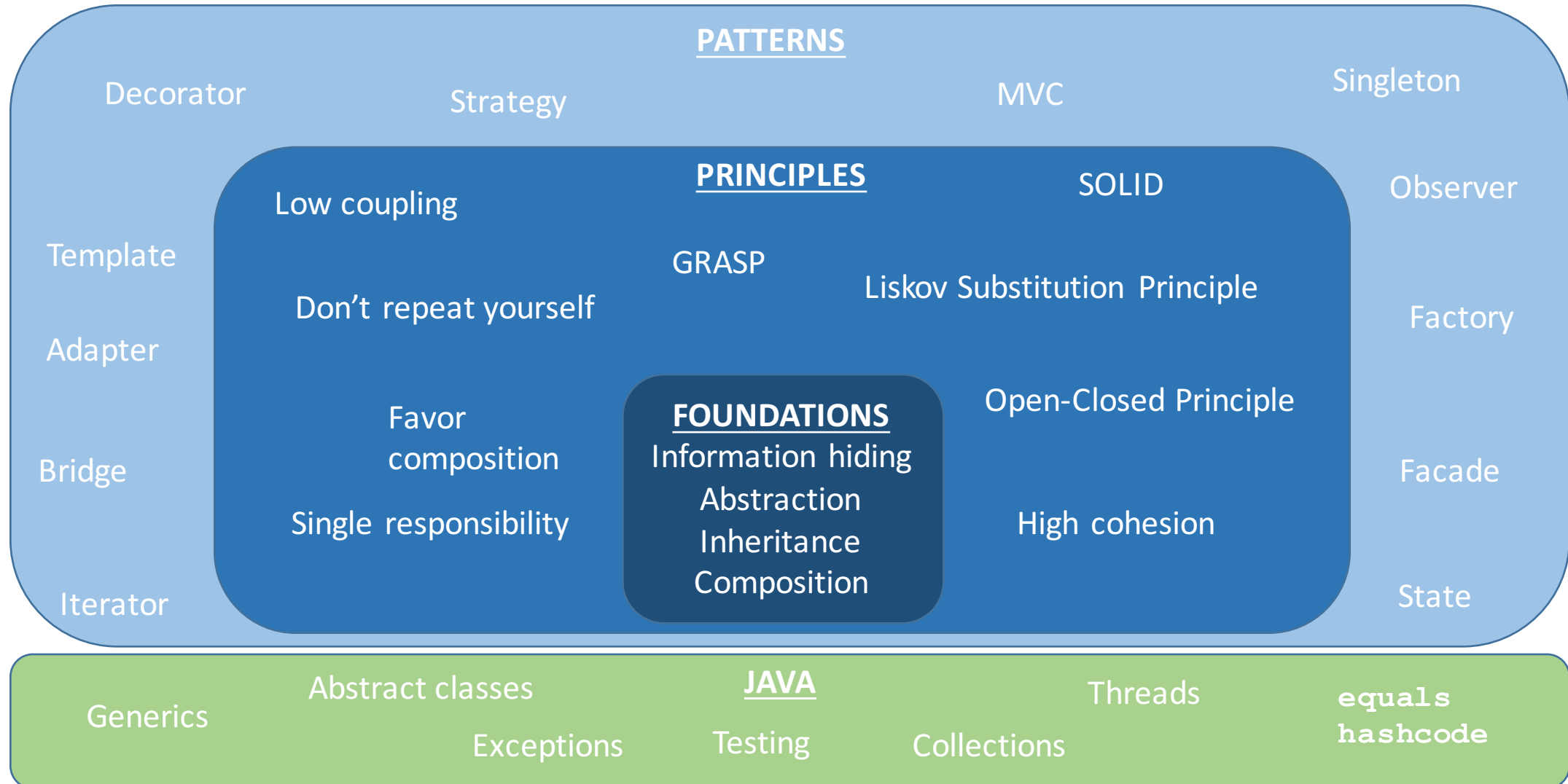


MUCH LATER...

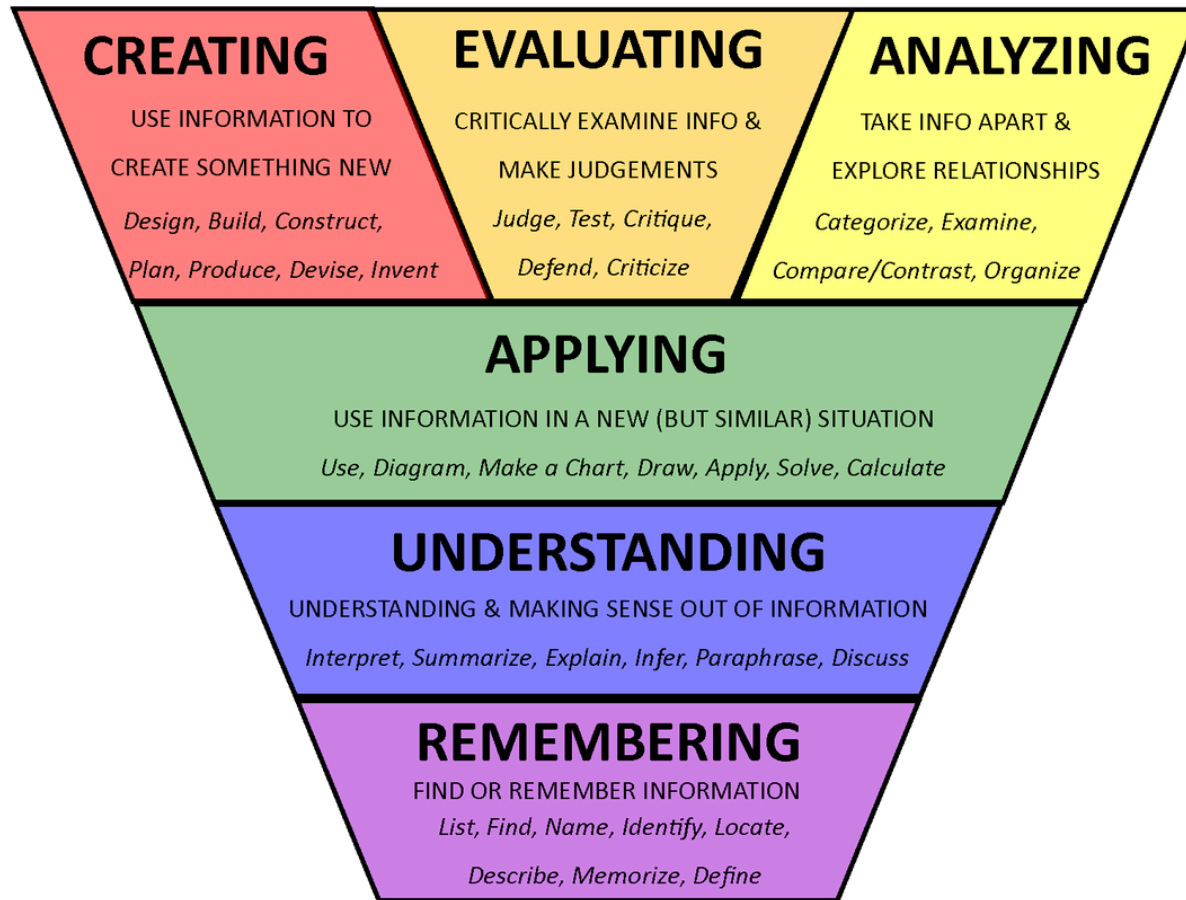
OH MY. I'VE
DONE IT AGAIN,
HAVEN'T I ?



Från verktyg till principer



Blooms lärandepyramid



TDA548:

Verktvg: → Applying

TDA551:

Verktvg: → Evaluating
Principer: → Applying

Kurshemsida

<http://www.cse.chalmers.se/edu/course/TDA551>

Kursens upplägg

- 2 ”moduler” per vecka, 7 veckor
- Modul = Övning -> laborationspass -> reflektion/föreläsning
 - Övning introducerar koncept, låter er arbeta med problemställning
 - Laborationspass låter er arbeta självständigt, med övningar och laborationer
 - En del laborationspass kommer att användas till specifika workshops
 - Föreläsning knyter ihop ämnet, visar på principer, diskuterar, reflekterar
- 3 obligatoriska laborationer
 1. Använda ett ramverk för att implementera spelet Snake (eller liknande)
 2. Bygga ett paket för geometriska former
 3. Analys och refaktorering av spelramverket från laboration 1
 - Muntligt redovisning
- Skriftlig tentamen – men information senare

Workshops/tutorials

- Första läsvecka börjar vi med några workshops under laborationspassen
- Genomföra tutorials, länkar hittar ni på kurshemsidan under tabben 'Resurser'
- Ämne
 - Dokumentation:javadoc
 - Enhetstestning: JUnit
 - Versionshantering: git
- Be handledarna om hjälp om ni fastnar
- Man skulle kunna växla mellan workshops och labben

Kursplan översikt (preliminär)

- Vecka 1: Översikt, recap, UML, testning
- Vecka 2-3: Arv (inheritance), typer och subtyper
- Vecka 4-5: Dependency reduction (med patterns)
- Vecka 6: State och immutabilitet (med patterns)
- Vecka 7: Diverse (exceptions, threads, design by contract, lambdas, ..?)
- Tentamen

Observera

- Kursen är (hyfsat) ny!
 - Materialet kan komma att förändras under kursens gång!
 - Hjälp till genom att påpeka områden som är oklara

Jag förväntar mig av er:

- Att ni är närvarande och deltar aktivt på övningar och föreläsningar
- Att ni använder laborationstiden konstruktivt
- Att ni deltar aktivt i grupparbetet kring laborationerna
- Att ni lämnar in/redovisar laborationerna i tid
- Då kommer kursen gå bra för er!

Saker ni kan sedan tidigare(?)

- Klasser
- Objekt
- Subklasser
- Primitiva typer
- Typomvandling
- Enum
- Interface
- Metoder
- Konstruktorer
- Call by value
- Call by reference
- Arv (inheritance)
- Polymorfism
- Association
- Inkapsling
- Synlighet
- Overloading
- Overriding
- Variabler
- Klassvariabler
- Scope
- Abstrakta klasser
- Abstrakta metoder
- Dynamisk bindning
- Arrayer
- If/While/For
- Enkel grafik

Grupper

- Använda samma grupper
- Niklas och jag är här för att lösa problem med grupper

Övning

- Ladda hem "DrawPolygons.java" från kursens hemsida
- Diskutera koden i grupp:
 - Vilka förändringar av koden skulle göra den bättre?
 - ... vad betyder "bättre"?
- Förbättra koden enligt de förändringar ni kommer fram till
- För mer utmaning:
 - Lägg till möjligheten att klicka ut nya polygoner
 - Hur bör denna förändring implementeras?