

EDA016 Programmeringsteknik för D

Läsvecka 15: Avslutning, Utblick

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2015

15 Avslutning, Utblick

- Att göra denna vecka
- Tenta
- Inlämningsuppgift
- Kursutvärdering
- Grumligtlådan
- Utblick
- Frågestund
- Om nästa läsperiod: Gäster Anna Axelsson & Martin Höst

Att göra i Vecka 15: Repetition, uppsamling, tentaplugg.

- 1 Uppsamling: **gör klart och redovisa kvarvarande labbar och inlämningsuppg. på veckans resurstider**
- 2 Träffas i samarbetsgruppen & hjälp varandra att tentaplugga.
- 3 Ingen labbtid denna vecka - använd resurstider för uppsamling och frågor inför tentapluggande.
- 4 Kolla på tentapluggtipsen på **förra föreläsningen**.

Extenta: Försättning på Sociala-nätverkstentan

Tenta: anmälning och anonyma tentor

- 1 Du **måste vara godkänd** på **alla** labbar + inlämningsuppg. för att få tenta.
- 2 Läs instruktioner och anmäl dig här:
- 3 Gruppbonus: Din grupptillhörighet och dess bonus meddelas via mejl. Kontakta mig direkt om något är felaktigt eller problematiskt.

Inlämningsuppgift Bank; diskussion + utblick

Exempel på klurighet

Specifikationen tillåter uttag som är större än saldot på kontot, men i det textuella användargränssnittet ska felmeddelande ges vid för stora uttag som inte medges.

Hur löser vi det?

Inlämningsuppgift Bank; diskussion + utblick

Exempel på klurighet

Specifikationen tillåter uttag som är större än saldot på kontot, men i det textuella användargränssnittet ska felmeddelande ges vid för stora uttag som inte medges.

Hur löser vi det?

Genom att implementera logik som förhindrar för stora uttag.

Inlämningsuppgift Bank; diskussion + utblick

Exempel på klurighet

Specifikationen tillåter uttag som är större än saldot på kontot, men i det textuella användargränssnittet ska felmeddelande ges vid för stora uttag som inte medges.

Hur löser vi det?

Genom att implementera logik som förhindrar för stora uttag.

- När man använder ett api behöver man förstå vad api:et omfattar och hur det är tänkt att användas.
 - Vad behöver jag själv kolla?
 - Vad kollas av koden jag använder?
 - Vem har ansvar för vad?
- Mer om design-frågor i OMD-kursen.

Inlämningsuppgift Draw; diskussion + utblick

Exempel på klurighet

Utvidgning: relativ förflyttning med tangentbordet.

Men klassen `l.th.cs.pt.shapes.Shape` har inga `GetX`, `GetY`

Hur löser vi det?

Inlämningsuppgift Draw; diskussion + utblick

Exempel på klurighet

Utvidgning: relativ förflyttning med tangentbordet.

Men klassen `se.lth.cs.pt.shapes.Shape` har inga `GetX`, `GetY`

Hur löser vi det? Två olika lösningar:

- Kopiera `se.lth.cs.pt.shapes.Shapes` till mitt eget paket och lägg till metoderna
- Eftersom attributen `x` och `y` är **protected** i `Shape` så kan vi med en ny klass `LocatableShape` **extends** `Shape` skapa en ny basklass för våra figurer och där implementera `GetX`, `GetY`

Inlämningsuppgift Draw; diskussion + utblick

Exempel på klurighet

Utvidgning: relativ förflyttning med tangentbordet.

Men klassen `se.lth.cs.pt.shapes.Shape` har inga `GetX`, `GetY`

Hur löser vi det? Två olika lösningar:

- Kopiera `se.lth.cs.pt.shapes.Shapes` till mitt eget paket och lägg till metoderna
- Eftersom attributen `x` och `y` är **protected** i `Shape` så kan vi med en ny klass `LocatableShape` **extends** `Shape` skapa en ny basklass för våra figurer och där implementera `GetX`, `GetY`

Viktiga generella observationer:

- Om du har tillgång till källkoden och får kopiera den kan du bygga vidare på koden (en s.k. **fork**), men vad göra om api:et kommer i ny version?
- Genom arv kanske det går att fixa det du saknar, men vad göra om api:et kommer i ny version?

CEQ

1 Om CEQ

Kursspecifik utvärdering om specifika kursmoment

- 1 Om kursspecifik utvärdering + länk till google forms

Grumligtlådan - avslutande diskussion och frågestund

Diskussion om några av begreppen ur grumligtlådan:

- **overloading**: flera metoder med samma namn
- **override**: ersätta metod i subklass
- **iterator**: ett objekt med metoder för att iterera över elementen i en samling
- "När du säger **Java** exakt vad menar du då?"
 - 1 Ett programspråk
 - 2 En virtuell maskin och exekveringsplattform
 - 3 En stor uppsättning standardbibliotek med färdiga klasser
 - 4 En global gemenskap av kompatibla öppenkällkodsbibliotek
- **volatile**: speciellt nyckelord för att skydd mot samtidig access av gemensamma variabler i jämlöpande trådar. Java har speciella språkkonstruktioner och bibliotek för att hantera trådar som exekverar samtidigt. Detta är ett avancerat och intressant område som ni läser mer om i realtidsprogrammering.

Utblick

Utblick: Vart är Java och JVM-teknologin på väg?

- En viktig trend är kombinationen av objekt-orientering och funktionsprogrammering och detta visar sig i **Java 8** med sina anonyma funktioner (s.k. lambda expressions) och nya samlingen `java.util.stream`.
- I **Java 9** som resultat av "**Project Jigsaw**" kommer mycket handla om modularisering av JDK som möjliggör skräddarsydda, mindre runtime-versioner av JVM.
- Mycket av utmaningarna handlar om att dra nytta av många kärnor och parallellprogrammering blir allt hetare. Därför jobbar man mer och mer med oföränderlig data som gör det lättare att dela upp beräkningar på flera kärnor.
- Det finns flera livaktiga språk som kör på JVM, t.ex. **Scala** och **Clojure** men också **Javascript** med skriptmotorn **Nashorn**
- Med kompilatorn för Scala till Javascript **Scala.js** kan man dela kod mellan en server som kör på JVM och en klient som kör i webbläsaren.

Utblick: Vart är software engineering på väg?

Några faktorer som skapar förändringstryck för ökad produktivitet i utvecklingsprocessen:

- Snabbast vinner.
- Jakt på kompetens.
- Sakernas internet.

Några starka trender:

- Open source
- Big Data
- Devops

Frågestund

Om nästa läsperiod: Gäster Anna Axelsson & Martin Höst

Ett stort TACK för...

- ... att ni kämpat så glatt!
- ... att ni ställt massor med frågor!
- ... att det har varit så hög närvaro på föreläsningarna!
- ... att ni är så konstruktiva och verkligen vill lära er!

Ett stort TACK för...

- ... att ni kämpat så glatt!
- ... att ni ställt massor med frågor!
- ... att det har varit så hög närvaro på föreläsningarna!
- ... att ni är så konstruktiva och verkligen vill lära er!

Ett stort LYCKA TILL på vägen till att bli en kompetent och innovativ systemutvecklare!