

Kursutvecklingsmöte pgk

Kickoff

Björn Regnell, Maj Stenmark, Gustav Cedersjö

Datavetenskap, LTH

11 mars, 2016

Agenda:

- 1 Introduktion
- 2 Grov tidplan
- 3 Studera Scala
- 4 Hur ska vi jobba?
 - Roller
 - Lunchmöten
 - Verktyg
 - Nätplatser
 - Repo
 - Avslutning: att göra härnäst

Välkommen!

Fantastiskt! att så många (ca 35) vill vara med och jobba med **kursutveckling** (VT2016) och/eller **undervisning** (HT2016) i nya kursen EDAA45 Programmering, grundkurs **pgk**

Ett nytt sätt att jobba:

Kursmaterial utvecklas open source av lärare **och** studenter

Upprop

- Vem är här? (Se länk till google spreadsheet i mejl)
- Namnbrickor

Grov tidplan

- I **mars** fokuserar vi på era studier i grunderna i Scala och studier av de verktyg vi kommer använda (latex, git, med flera). De som är DevLead börjar planera/organisera arbetet med labbarna under mars.
- Från **april** startar utveckling & test av labbar, parallellt med fördjupade Scala-studier. Dessa saker ska utvecklas och testas:
 - 1 en handledning per lab
 - 2 ett givet kodskellett per lab i studenternas workspace
 - 3 hemlig lablösning för lärare: Obs! får ej delas öppet!!
 - 4 mönsterlösningar till övningarna i kompendiet
- Senast i slutet av **maj** ska alla labbarna vara färdiga och testade och alla övningarna ha mönsterlösningar.

Studera Scala

- Vi använder denna gratis nätbok:
<https://www.artima.com/pins1ed>
- Ni läser själva i förväg
- Alla hjälps åt på lunchmöten att förklara och förstå
- Bilda gärna studiegrupper att jobba i
- Länk till community-dokumentation:
<http://www.scala-lang.org/documentation/>
- Länk till scaladoc:
<http://www.scala-lang.org/api/current/>

Om pins1ed

Första upplagan <https://www.artima.com/pins1ed>

- Ingen nybörjarbok: förutsätter kunskap om OO och Java.
- Täcker bara fram till Scala 2.8, men det gör inte så mycket då vi koncentrerar oss på grunderna och inte det mest avancerade eller nyaste
- Saker som vi kan stöta på som kommit sedan Scala 2.8: App, implicit classes, value classes, extension methods, string interpolators, additional collection methods, Try, import scala.language._
- Länk: [språkhistorik](#)

Hemläxor – tidplan scala-studier kommande veckor

- Hemläxa till mån lunchmöte 14/3:
Läs kapitel 1-2. Gå vidare nedan om du har tid/ork.
- Hemläxa till tors lunchmöte 17/3:
Läs kapitel 3-4 i pins1ed. Börja testa övn1.
- Hemläxa till mån lunchmöte 21/3:
Läs kapitel 5-6 i pins1ed. Börja testa lab1 med Kojo.
- Hemläxa tors lunchmöte 24/3:
Läs kapitel 7-8 i pins1ed. Test av övn1 och lab 1 klart.

OBS! Håll reda på hur lång tid det tar att göra övn1 och lab1.
Jag undrar om de är lagom omfattande för "medelstudeneten".
Länk till kompendium i pdf (uppdateras kontinuerligt):

<https://github.com/lunduniversity/introprog/raw/master/compendium/compendium.pdf>

Roller

■ DevLead

- övergripande kvalitets- och leveransansvar
- utveckla handledning i latex i kompendiet
- utveckla kodskellett i studentworkspace
- utveckla hemlig lösning
- samordna insatser från TestLead och Dev

■ TestLead

För labbar:

- testa labhandledning
- testa kodskelett i workspace
- samordna insatser från Test.

För övningar:

- utveckla publik lösning i latex i kompendiet
- samordna insatser av Test.

■ Dev

- Hjälpa DevLead med det som är lämpligt att delegeras

■ Test

- Hjälpa TestLead med det som är lämpligt att delegeras

Lunchmöten mån + tors

- Vi kommer att ha en stående lunchmötestid måndagar och torsdagar, med start nu på måndag kl 12:15 alltid i E:2116.
- **Boka in så många mån/tor-luncher du kan till 16 juni.**
- På lunchmötena hjälps vi åt att identifiera saker att lära mer om och vi varvar plenumdiskussioner med genomgångar och arbete i grupp.
- Mötesledaren (oftast någon av Gustav, Maj, Björn, ibland ngn av er) kommer styra upp det hela och ibland ge korta föredrag el. lajvkodning som förklarar begrepp i Scala.
- Så småningom kommer vi även använda lunchmötena för att synka arbetet med att utveckla labbarna och övningarna.

Verktyg vi ska använda

- terminalverktyg, t.ex.:
scala, scalac, scaladoc, java, javac, jar, javadoc
- din favoriteditor, t.ex gedit
- Kojo: en lättanvänd IDE för Scala
- ScalalIDE: en eclipse-plugin för Scala
- git för versionshantering
- GitHub för öppen central kodlagring
- GitLab för hemlig central kodlagring
- sbt: scala build tool för att bygga (autoladda jarfiler, automatisera kompilering och exekvering)
- latex för labbhandledning och övningsmönsterlösningar

Verktyg vi ska använda

- terminalverktyg, t.ex.:
scala, scalac, scaladoc, java, javac, jar, javadoc
- din favoriteditor, t.ex gedit
- Kojo: en lättanvänd IDE för Scala
- ScalalIDE: en eclipse-plugin för Scala
- git för versionshantering
- GitHub för öppen central kodlagring
- GitLab för hemlig central kodlagring
- sbt: scala build tool för att bygga (autoladda jarfiler, automatisera kompilering och exekvering)
- latex för labbhandledning och övningsmönsterlösningar

Hur många vill redan snart jobba på skolans datorer?

Var? Nätplatser där vi jobbar

- **"upstream"** på GitHub
<https://github.com/lunduniversity/introprog>
- Listor med utsedda roller, mgmt, etc. i Google docs
"pgk-work" (Se länk till google spreadsheet i mejl)
- **"closed upstream"** i GitLab
hemliga lösningar, grejer bara för lärare
<https://git.cs.lth.se/bjornregnell/pgk>

Hur jobba med git och GitHub: 1) Sätta upp repo

Det centrala repot <https://github.com/lunduniversity/introprog> kallas **upstream**

- 1 Installera git på din egen dator
- 2 Registrera användare på GitHub om du inte redan gjort det.
Tips: tänk efter noga vilket användarnamn du ska ha. Förslag om det inte redan är upptaget: fornamnefternamn i ett ord utan versaler och konstiga tecken.
- 3 Installera GitHub-klienten på din dator om du kör windows eller mac
<https://desktop.github.com/>
Kör du linux: använd terminalen eller Atlassian SourceTree
- 4 Om du är DevLead: Gör **fork** på GitHub av
<https://github.com/lunduniversity/introprog>
- 5 Om du är Dev:
Be din DevLead tillåta din användare att pusha till hens fork på GitHub.
- 6 Klona DevLeadens fork till din egen dator.
Då får du en lokal kopia som du jobbar i.

Hur jobba med git och GitHub: 2) Arbetsgång

Det är **viktigt** att hålla forken uppdaterad mot upstream.

Om man ändrar i föråldrade filer kan det bli onödiga **merge-konflikter**.

- 1 Synka din fork
<https://help.github.com/articles/syncing-a-fork/>
- 2 Ändra i de filer som ingår i ditt ansvarsområde enligt överenskommelse med DevLead.
- 3 Gör `git commit -am "msg"` (bekräftar din ändringar lokalt)
- 4 Gör `git pull` (hämtar ev ändringar från forken på GitHub)
- 5 Gör `git push` (skickar dina ändringar till forken på GitHub)
(I stället för 3-5 ovan kan du klicka "synch" i *GitHub-klienten*.)
- 6 Om du är DevLead:
När en dellösning är färdig gör en **new pull request** på GitHub.

Att göra nu

- 1 Fyll i vad du vill jobba med i google docs spreadsheet pgk-work, i fliken "work" kolumnerna E,F,G
- 2 Anmäll ev. intresse för DevLead, TestLead i google docs spreadsheet pgk-work, i fliken "lead" längre ner i högerkolumnerna från C28 & D28 (skriv inte över andras...)
- 3 Fyll i nuvarande kunskapsnivå i google docs spreadsheet pgk-work, i fliken "learn"
- 4 Börja på hemläxorna: läs minst kap 1-2 i helgen
- 5 Kom på måndag lunch till E:2116 med frågor om allt möjligt men mest på kap 1-2
- 6 Boka in detta event den 20 April kl 10-12 och 18-21:
<http://cs.lth.se/scala20160420>

Frågor??