INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

DIT272 Formella metoder i mjukvaruutveckling, 7,5 högskolepoäng

Formal Methods in Software Development, 7.5 credits

Avancerad nivå / Second Cycle

Fastställande

Kursplanen är fastställd av Institutionen för data- och informationsteknik 2020-12-18 att gälla från och med 2021-08-30, höstterminen 2021.

Utbildningsområde: Naturvetenskapligt 100 %
Ansvarig institution: Institutionen för data- och informationsteknik

Inplacering

Kursen ingår i Computer Science Master's Programme och ges även som fristående kurs vid Göteborgs universitet.

Kursen kan ingå i följande program: 1) Datavetenskapligt program (N1COS) och 2) Computer Science, Master's Programme (N2COS)

Huvudområde	*Fördjupning*
Datavetenskap	A1F, Avancerad nivå, har kurs/er på avancerad nivå som förkunskapskrav

Förkunskapskrav

Förkunskapksravet är avklarade kurser om 120 hp i ämnet Datavetenskap, eller motsvarande, särskilt DIT201 Logic in Computer Science, 7,5 hp, och en 7,5 hp kurs i objektorienterad programmering (eller motsvarande) är ett krav.

Följande kunskapsnivå i Engelska krävs; Engelska 6/Engelska B eller motsvarande från ett erkänt internationellt test, t.ex. TOELF, IELTS.

Lärandemål

Efter godkänd kurs ska studenten kunna:

Kunskap och förståelse

- redogöra möjligheter och begränsningar hos logikbaserade verifikationstekniker för att bedöma och förbättra mjukvarukvalitet,
- avgöra vad som kan och inte kan uttryckas i en given formalism för specifikation eller modellering,
- avgöra vad som kan och inte kan analyseras med en given logik och bevismetod,

Färdigheter och förmåga

- formellt uttrycka säkerhetsegenskaper och liveness hos (parallella) program,
- beskriva grunderna i verifikation av säkerhetsegenskaper och liveness med hjälp av modellkontroll (model checking),
- använda verktyg som automatiserar modellkontroll av säkerhetsegenskaper,
- skriva formella specifikationer för klasser i objekt-orienterade program med hjälp av kontrakt och klassinvarianter,
- beskriva hur förhållandet mellan program och formell specifikation kan representeras i en programlogik,
- verifiera funktionella egenskaper hos enkla Javaprogram med ett verifikationsverktyg,

Värderingsförmåga och förhållningssätt

- bedöma och kommunicera betydelsen och vikten av korrekthet i mjukvaruutveckling,
- lösa problem relaterade till utveckling av välfungerande mjukvara genom abstraktion, modellering och rigorösa resonemang.

Innehåll

Kursens syfte är att lära ut kunskap, teknik och omdöme angående viktiga tekniker inom formella metoder: modellkontroll (model checking) och deduktiv verifikation. Båda stilarna introduceras på tre sätt: konceptuellt, teoretisk och praktiskt, genom användning av ett specifikt verktyg. Kursen bygger på kunskap om första ordningens logik och temporallogik, och visar hur dessa formalismer kan appliceras, och utökas för verifikation av mjukvara.

För modellkontroll täcker kursen följande ämnen:

- ett specifikationsspråk för parallella processer,
- verifiering av påståenden,
- synkronisering,
- verifikation av säkerhets- och livenessegenskaper som är skrivna i temporal logik.

För deduktiv verifikation täcker kursen följande ämnen:

- ett specifikationsspråk på enhetsnivå för Java-program,
- en logik för verifikation av Java-program,
- verifikation av Java-program, i meningen att implementationen av en enhet uppfyller dess specifikationen.

Delkurser

1. **Muntlig Tentamen** *(Oral Examination),* 5 hp
   Betygsskala: Mycket väl godkänd (5), Väl godkänd (4), Godkänd (3) och Underkänd (U)
2. **Laboration** *(Laboratory),* 2,5 hp
   Betygsskala: Godkänd (G) och Underkänd (U)

Former för undervisning

Det är cirka två föreläsningar per vecka och det finns en övning per vecka. Studenterna utför praktiska exempelövningar med hjälp av verktygen i laborationer.

Undervisningsspråk: engelska

Former för bedömning

Kursen examineras genom en muntlig tentamen (U-VG) vid slutet av kursen och obligatoriska inlämningsuppgifter som lämnas in under kursens gång (U-G). Inlämningsuppgifterna görs vanligtvis i par.

Om student som underkänts två gånger på samma examinerande moment önskar byte av examinator inför nästa examinationstillfälle, ska sådan begäran inlämnas skriftligt till kursansvarig institution och bifallas om det inte finns särskilda skäl däremot (HF 6 kap § 22).

I det fall en kurs har upphört eller genomgått större förändringar ska studenten i normalfallet garanteras tillgång till minst tre provtillfällen (inklusive ordinarie provtillfälle) under en tid av åtminstone ett år med utgångspunkt i kursens tidigare uppläggning.

Betyg

På kursen ges något av betygen Mycket väl godkänd (5), Väl godkänd (4), Godkänd (3) och Underkänd (U).
På kursen ges något av betygen Underkänd (U), 3, 4 eller 5.

Kursen bedöms genom två inlämningsuppgifter som normalt genomföras i grupper om två, samt en muntlig tentamen vid kursens slut. Inlämningsuppgiter och muntlig tentamen kan godkännas oberoende av varandra. För att få godkänt på hela kursen

krävs godkänt på både inlämningsuppgifterna och den muntlig tentamen. På den muntlig tentamen ges något av betygen Underkänd (U), 3, 4 eller 5. Betyg för godkända studenter avgörs av tentamensresultatet.

Kursvärdering

Kursen utvärderas genom möten både under och efter kursen mellan lärare och studentrepresentanter. Därutöver används en anonym enkät för att få skriftlig information. Resultatet av utvärderingen används för att förbättra kursen genom att visa på delar som kan läggas till, förbättras, ändras eller tas bort.

Övrigt

Kursen ersätter DIT270 Software Engineering using Formal Methods. Den här kursen kan inte ingå i en examen som innehåler DIT270. Den kan inte heller ingå i en masterexamen som bygger på en annan bachelorexamen där DIT270 ingår.

Kursen är en samläst kurs med Chalmers.

Kurslitteratur kommer att publiceras senast 8 veckor innan kursstart.