

KURS **Objektorienterad programmering, 7.5 p**
KURSBETECKNING DAT042
LÄSÅR 2012/2013, lp 1
FÖR D2

EXAMINATOR Uno Holmer, tel. 772 5730, DIT plan 6, rum 6455, (e-post enl. ovan)
FÖRELÄSARE ” ”

ÖVNINGSLEDARE
och **HANDLEDARE** Cecilia Kjellman, [cecilia.kjellman 'at' gmail.com](mailto:cecilia.kjellman@gmail.com)
Niklas Lönnerfors, [niklas_lonnerfors 'at' yahoo.se](mailto:niklas_lonnerfors@yahoo.se)
Karl Schmidt, [karsch 'at' student.chalmers.se](mailto:karsch@student.chalmers.se)

KURSREPRESENTANTER
Filip Bertilsson, [filipbe 'at' student.chalmers.se](mailto:filipbe@student.chalmers.se)
Gustav Davidsson, [gusdavi 'at' student.chalmers.se](mailto:gusdavi@student.chalmers.se)
Erik Ivarsson, [ierik 'at' student.chalmers.se](mailto:ierik@student.chalmers.se)
Simon Planhage, [planhage 'at' student.chalmers.se](mailto:planhage@student.chalmers.se)
Dean Todevski, [todevski 'at' student.chalmers.se](mailto:todevski@student.chalmers.se)

FÖRKUNSKAPSKRAV

Genomgången kurs som behandlar elementära programmeringsbegrepp, motsvarande TDA605 introduktion till programmering. Kunskaper i grundläggande imperativ programmeringsteknik motsvarande EDA480 maskinorienterad programmering. Genomgången kurs som behandlar grundläggande funktionell programmering, motsvarande TDA555 introduktion till funktionell programmering. Genomgången kurs som behandlar diskret matematik, motsvarande TMV210 inledande diskret matematik.

SYFTE

Kursens syfte är att lära ut principerna för det objektorienterade synsättet och hur man konstruerar objektorienterade program. Det objektorienterade synsättet har idag en mycket stark ställning när det gäller programutveckling. Program som är uppbyggda på objektorienterat sätt, består av ett antal samverkande delar som kallas objekt. Objekten kapslar in data och erbjuder metoder för att avläsa och uppdatera dem. Objekten beskrivs av klasser.

LÄRANDEMÅL

Efter fullgjord kurs skall studenten kunna:

- självständigt konstruera objektorienterade program som både använder sig av egenhändigt konstruerade klasser och av färdiga standardklasser.
- konstruera händelsestyrda program med grafiska användargränssnitt.
- använda standardklasser för bl.a. datasamlingar, strömmar, filer och trådar.
- söka information om standardklasser i java API.
- dokumentera program med JavaDoc.
- använda några vanligt förekommande kodutvecklingsverktyg.
- genomföra testning av program med JUnit.
- beskriva enkla objektorienterade modeller i ett objektorienterat modelleringspråk.
- använda några olika designmönster.

INNEHÅLL

I kursen används programspråket Java och modelleringsspråket UML. Följande moment behandlas:

- Grundläggande begrepp i objektorienterad programutveckling. Skillnader mellan de objektorienterade och funktionsorienterade synsätten.
- Objekt och klass, datainkapsling och informationsgömning.
- Konstruktörer, metoder, instansvariabler, klassvariabler.
- Modularisering av program.
- Användning av dokumentation för standardbibliotek.
- Användning av standardklasser för datasamlingar.
- Dokumentation av egna program med Javadoc. Kodningsstandard, namnsättning och kommentering.
- Testning av program. Automatisk regressionstestning med JUnit och verktygsstöd.
- Arv, dynamisk bindning och polymorfism.

- Abstrakta klasser och gränssnitt.
- Grafiska användargränssnitt. Händelser och lyssnare.
- Model-View-Controller-arkitekturen för program med grafiska användargränssnitt.
- Strukturerad felhantering.
- Strömmar och filer.
- Något om aktiva objekt och trådar.
- Likhets- och olikhetsrelationer mellan objekt. Hashkoder. Algebraiska egenskaper och implementering i Java.
- Grund och djup kopiering av objekt.
- Orientering om programutvecklingsprocesser och modellering i UML.
- Några vanliga designmönster, t.ex. Singleton, Decorator, Composite, Factory method, Observer.

KURSAKTIVITETER

Undervisningen består av föreläsningar, lärarledda gruppövningar, samt handledda laborationer i datasal med **obligatorisk inlämning**. Kursen kräver aktivt och regelbundet arbete med laborationer och övningsuppgifter. *Du kommer sannolikt att behöva ägna betydligt mer tid åt uppgifterna än de schemalagda/handledda tiderna.*

Den första laborationen (av totalt fyra) redovisas **senast fredagen den 7/9**
– alltså redan i slutet av den första läsveckan!

LITTERATUR

Det rekommenderas att skaffa en bok som täcker grundläggande objektorienterad programmeringsteknik med programmeringsspråket Java. För den som avser att senare i utbildningen läsa fler programmeringskurser där Java används, kan det vara en fördel om boken förutom grunderna även omfattar parallellprogrammering med trådar, filhantering, samt kommunikation. Ett exempel på en sådan bok är *Java direkt* av Jan Skansholm. Boken brukar finnas hos Cremona. Andra böcker med motsvarande innehåll kan givetvis också användas, t.ex. *Big Java* av Horstmann. En mycket pedagogisk och lättillgänglig bok för den som tycker att programmering är svårt är *Objects First with Java*, av Barnes & Kölling. Här betonas principer för god design av objektorienterade program mer än Java-detaljer och boken fungerar därför kanske mindre bra som Java-referens än de ovan nämnda.

Mängden information om Java på Internet är i det närmaste outtömlig. Kurshemsidan innehåller ett litet urval länkar till javadokument samt några gratisverktyg som du kan ladda ner och installera på din egen dator. En mycket viktig referens vid det praktiska programmeringsarbetet är Java API som innehåller dokumentationen om Javas standardklasser. Ha alltid dessa sidor i webbläsaren under laborationerna, och lär dig att hitta i dem! Och sist, men inte minst, är det väl som vanligt ofta enklast att hitta det man söker med en viss sökmotor som börjar på G.

PM, OH, m.m. Allt material distribueras via kursens webbplats (se nedan).

PROGRAMUTVECKLINGSMILJÖ

BlueJ, Jgrasp, eclipse eller motsvarande under valfritt operativsystem. BlueJ är en utvecklingsmiljö som är speciellt avsedd för inläring av objektorienterad programmering med Java. Den är okomplicerad och enkel att komma igång med och rekommenderas starkt som första verktyg. eclipse är ett mer fullskaligt verktyg för avancerad systemutveckling som används mycket i näringslivet. Det rekommenderas inte för nybörjare, men kan vara intressant att bekanta sig med senare i kursen. Laborationer redovisas via Fire-systemet, se nedan.

KURSWEBB

Kursmaterialet kan hämtas på kursens webbplats som nås via Chalmers studieportal eller via föreläsarens hemsida www.cse.chalmers.se/~holmer.

REDOVISNING AV LABORATIONER

Laborationerna genomförs i grupper om två personer och redovisas elektroniskt via det webbaserade Fire-systemet. *För att kunna använda Fire måste varje labbgrupp registrera sig i systemet.* Se kursens hemsida för mer detaljerad information om labbredovisning, där finns också en länk till registreringsidan för Fire.

EXAMINATION

Tentamen	Skriftlig tentamen i TP 1 omfattande 4.5 p.
Laborationer	I kursen ingår 4 obligatoriska laborationer som tillsammans ger 3 p. Senaste redovisningsdatum finns i Fire-systemet. <i>Högst två returer ges på varje laborationslösning och du har max två veckor på dig att rätta felen efter varje retur.</i> <i>Laboration eller rättad version som inte lämnas in i tid underkänns, likaså rättad version som inte gjort väsentliga framsteg efter returen.</i>
Slutbetyg	Slutbetyg i skala (3-5) ges efter godkända laborationer och godkänd tentamen och baseras på tentamensbetyget.
Hjälpmedel vid tentamen	Java API, ett papper med ett urval av några vanliga javaklasser. Bifogas tentamenstesen och finns även på kurshemssidan.

SCHEMATID Aktuellt schema finns i Chalmers studieportal.

KURSPLANERING

Föreläsningarna kommer att belysa objektorienterad programmering på ett principiellt plan, bl.a. genom många exempel. Koden för dessa kommer att finnas tillgängliga för nedladdning på kurshemsidan. Föreläsningarna avser inte att behandla alla detaljer i språket java. Kursen startar alltså inte med en genomgång av all grundläggande javasyntax, utan fokuserar mer på att tidigt förmedla ett objektorienterat synsätt. Litteraturreferenserna i tabellen nedan avser avsnitt i Skansholms bok (6:e uppl.) som helt eller delvis täcker innehållet i respektive föreläsning. Kapitel 7 om grafiska komponenter, 15 om bilder och ljud, samt 18 som behandlar kommunikation får betraktas som överkurs.

V.	Dag	Nr	Föreläsning	Litteratur
36	mån	1	Introduktion till kursen och OOP	S 1
	tis	2	Objekt och klass, instansvariabler, klassvariabler, metoder	S2-3
	tor	3	Objektinteraktion, referenser, listor	S2.5, S4.5.1-2, S9.7
	fre	4	Fält, iteration, iteratorer	S3.8-9, S9.1-3, S9.8-9, S17.4-5
37	mån	5	Associativa samlingar, konstanter	S17.7.1-3, S3.3, S2.3
	tis	6	Testning, enhetstestning med JUnit	<i>mtrl utdelas</i>
	fre	7	Arv	S10.1-3, S10.5
38	mån	8	Polymorfism	S10.4, S10.6
	tis	9	Abstrakta klasser och gränssnitt	S10.7-8
	fre	10	Programdokumentation, kompilering av javaprogram	S19.1, S1.1-5, S9.4
39	mån	11	Grafiska användargränssnitt	S6, S12.1-9, S14
	tis	12	Grafiska användargränssnitt forts.	S8
	fre	13	MVC-modellen, observer-mönstret	<i>mtrl utdelas</i> , S12.10
40	mån	14	Strukturerad felhantering	S11
	tis	15	Filer och strömmar	S16
	fre	16	Aktiva objekt och trådar	S13
41	mån	17	Kopiering av objekt, serialisering, objektströmmar	<i>mtrl utdelas</i> , S10.10.4, S16.4.2
	tis	18	Objektrelationer. Likhet, ordningar och hashkoder.	S10.10.1-3, S10.11
	fre	19	Applikationsdesign, utvecklingsprocess, modellering	S4
42	mån	20	Designmönster, Singleton, Abstract factory m.fl.	S sid. 186, 341
	tis		<i>repetition</i>	
	fre		<i>reserv</i>	
43			Tentamen: tor 25/10 f 2012	