

Skolan för Datavetenskap och kommunikation

PROGRAMMERINGSTEKNIK

FÖRELÄSNING 14

INFORMATION OM P-UPPGIFTEN

- P-uppgiften
- Spec
- Prototyp
- Granskning
- Krav på P-uppgiftslösningen
- Redovisning av P-uppgiften

P-UPPGIFTEN:TIDER

- Fyra obligatoriska delmoment:

Spec	redovisas på övning	29/3
Prototyper (4 st)	redovisas på labbarna	30/3, 6/4, 13/4, 20/4
Granskning	med kurskamrat	21/4
Slutredovisning	på bokad tid	26/4 och 29/4

OM P-UPPGIFTEN

- P-uppgiften är ett **individuellt** moment.
- Du får inte samarbeta med någon annan.
- Läs **hederskodex** igen innan du börjar.
- Varje grupp har en mapp med uppgifter att välja bland.
- Du måste välja ur din egen grupps mapp eller hitta på en egen (maila i så fall beskrivning till Linda senast fredag 11/3).

SPECEN

- Specen ska läggas in i Dropbox och redovisas muntligt på övningen 29/3
- Specen ska beskriva datastrukturer, funktioner, klasser (med attribut och metoder) och algoritm.
- Genomgång (med exempel) på övningen idag
- Specen kan ge upp till 4 bonuspoäng

PROTOTYP

- En prototyp är en körbar, men inte färdig, version av ditt program.
- Tanken är att du ska
 - få in rutinen att bygga ut programmet successivt
 - testköra varje ny version.
- Prototyper kan ge upp till 4 bonuspoäng (en poäng per prototyp)

GRANSKNING

- Innan det färdiga programmet kan redovisas ska det testas (granskas) av en kurskamrat.
- Din granskare ska kritiskt granska ditt program, testköra det och fylla i ett granskningsprotokoll (kommer att gås igenom på övning).
- Syftet med granskningen är att
 - Den du granskar får tips om förbättringar
 - Du ser ditt eget program med nya ögon
 - Du lär dig att kvalitetsbedöma kod

KRAV PÅ P-UPPGIFTEN

- För att du ska bli godkänd måste ditt program uppfylla vissa krav, t ex att det ska vara användarvänligt och lättläst.
- Läs mer på kurswebbsidan

P-uppgiften

Krav på P-uppgiftslösningen

EXEMPEL PÅ KRAV

- Programmet ska vara kommenterat upptill med författare och datum.
- Programmet ska vara användarvänligt.
- Programmet ska vara uppdelat i lagom långa funktioner, och klasser.
- Namn på variabler och funktioner ska vara vettiga.
- Nästan identiska kodstycken ska inte upprepas. Gör i stället generella funktioner.
- Varje variabel och funktion/metod ska vara kommenterad.

REDOVISNING

- Boka tid för redovisning på webben.
- Ta med följande till redovisningen:
 - Uppgiftslydelsen
 - Specen
 - Programmet
 - Granskningsprotokollet och granskaren
- Om du redovisar på egen dator ska den vara uppkopplad mot Internet, eftersom vi (för att förhindra fusk) jämför din lösning mot alla tidigare lösningar.

HUR GÅR DET TILL?

- Du får intyga (signera) att du följt hederskodexen
- Assen kollar leg
- Granskaren får berätta
- Assen provkör, kollar koden, frågar
- Assen fyller i sitt protokoll
- Assen kollar att programmet inte är plagierat
- Du får betyg (A,B,C,D,E eller komplettering)

VAD KOLLAR ASSEN?

- Att programmet gör det som står i uppgiftslydelsen.
- Att programmet uppfyller övriga krav på P-uppgifter.
- Att du kan redogöra för alla detaljer i programmet.
- Att du kan motivera designval.

BETYG E

för betyg E ska du visa att du kan:

- följa reglerna i programspråkets syntax,
- tillämpa och redogöra för regler för god programmeringsstil (såsom användarvänlighet, kommentarer, felhantering, strukturering, flexibilitet),
- upptäcka och korrigera programmeringsfel,
- modifiera givna program,
- överföra data mellan fil och program,
- identifiera behovet av och använda styrstrukturer (villkorssatser och slingor),

BETYG E (FORTS)

- dela upp ett större problem i hanterliga delar och konstruera funktioner för dessa,
- använda de datastrukturer som finns inbyggda i programspråket, samt välja datastrukturer som passar för det aktuella problemet,
- *använda och konstruera egna klasser,*
- granska andras program

BETYG D

för betyg D ska du dessutom visa att du kan:

- följa en given tidsplan för arbetet,
- skriva ett perfekt program som i alla avseenden uppfyller givna krav på användarvänlighet, begriplighet och strukturering
- *Kraven på ett perfekt program gäller hela programmet, inklusive extrauppgifter för C, B resp A*

BETYG C

för betyg C ska du dessutom visa att du kan:

- infoga felhantering för att få ett program som inte kraschar för felaktiga indata

BETYG B

för betyg B ska du dessutom visa att du kan:

- konstruera och implementera en svårare algoritm

BETYG A

för betyg A ska du dessutom visa att du kan:

- sätta dig in i och använda en större modul (t ex tkinter, pygame, urllib) för att utveckla ditt program

BONUSPOÄNG

- Om du har 35 bonuspoäng eller mer kan du höja ditt betyg ett steg från E, D, C eller B.
- Bonuspoängen gäller ett år från kursstart.

PLUSSA

- Om du redovisar grunduppgiften och får minst E innan kursen är slut har du möjlighet att plussa till högre betyg.
- Plussningen måste göras senast i januari 2017.

SPEC - STRATEGI

1. Hur ska det se ut på skärmen när du kör ditt program?
2. Vilka data använder ditt program? Skriv upp exempel!
3. Vilka datastrukturer behöver du? Rita!
4. Vad ska programmet göra? Låt varje verb bli en funktion.
5. Vilka funktioner opererar på datastrukturen?
6. Kan du kombinera data och funktioner till en klass?
7. Behöver du filer? Vad ska dom innehålla?
8. Vad är det första ditt program ska göra?
9. I vilken ordning ska saker hända i ditt program?

EXEMPEL

Fiktiv P-uppgift:

Skriv ett program som hanterar hotellbokningar.

Välkommen till hotell Kudden!

Vill du ha ett dubbelrum eller enkelrum? *Dubbelrum*

Ankomstdatum? *20160307*

Antal nätter? *4*

Det finns ett ledigt dubbelrum. Vill du boka? *Ja*

Namn: *Ludde Dunér*

Du har bokat ett dubbelrum med ankomst 20160307 och avresa
20160311



DATA

data	exempelvärde
antal dubbelrum	7
antal enkelrum	3
rumsnummer	25
antal sängar	2
ankomstdatum	20160307
antal dagar	4
avresedatum	20160311
gästens namn	Ludde Dunér
gästens epost	ludde@dmil.se
gästens kreditkortsnummer	1234555512345555

HUR HÄNGER DATA IHOP?

data	
antal dubbelrum	} <i>värden för detta hotell</i>
antal enkelrum	
rumsnummer	} <i>olika värden för varje rum objekt Rum</i>
antal sängar	
ankomstdatum	} <i>olika värden för varje bokning objekt Bokning</i>
antal dagar	
avresedatum	
gästens namn	
gästens epost	
gästens kreditkortsnummer	

HUR SKA DATASTRUKTUREN SE UT?

Flera olika möjligheter, t ex

- Dictionary med datum som nyckel.
- Lista med rummen.
- Lista med bokningarna.

Här finns inget "rätt svar" - välj det du tycker verkar praktiskt.

RITA DATASTRUKTUREN MED DATA

Bokning 0

- Rum 11
- rumstyp *enkelrum*
- ankomst 20160401
- avresa 20160403
- gästnamn *Ele Fint*
- epost *ele@fint.se*
- kortnr 1234567812345678

Bokning 1

- Rum 21
- rumstyp *dubbelrum*
- ankomst 20160307
- avresa 20160311
- gästnamn *Ludde Dunér*
- epost *ludde@dmail.se*
- kortnr 1234567822225678

Bokning 2

- Rum 22
- rumstyp *dubbelrum*
- ankomst 20160401
- avresa 20160403
- gästnamn *Ele Fint*
- epost *ele@fint.se*
- kortnr 123456782446678

VAD SKA PROGRAMMET GÖRA?

- Fråga gästen om önskat datum mm
- Läs in tidigare bokningar från fil
- Kolla om ledigt rum finns
- Visa gästen erbjudande om rum
- Be gästen om namn, epost, kortnummer
- Lägg in ny bokning i datastrukturen
- Skriv ut bekräftelse av bokningen
- Spara bokningar på fil

FUNKTIONER

Varje sak programmet ska göra får bli en funktion. Exempel:

funktion guestSearch

"" Fråga gästen om ankomstdatum, antal nätter, typ av rum ""

Indata till funktionen (parametrar): -

Utdata från funktionen (returvärden): ankomstdatum, antal nätter,
typ av rum

METODER

Vissa funktioner kan passa bättre som metoder. Fundera över vilka data funktionen använder. Exempel:

Skriv ut bekräftelse av bokningen

skulle passa bra som metod i klassen Bokning eftersom den ska skriva ut data från bokningen.