

## 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 vecka 15
  - Prototyper - läggs upp i Dropbox
  - Granskning – en vecka innan du redovisar
  - Slutredovisning
    - senast vecka 19 för CLGYM/Lingvistik
    - senast vecka 20 för CSAMH

## 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).

## SPECIFIKATION

- Specen ska läggas in i Dropbox och redovisas muntligt på övningen vecka 15.
- Specen ska beskriva datastrukturer, funktioner, klasser (med attribut och metoder) och algoritm.
- Genomgång (med exempel) på övningen vecka 14
- Specen kan ge upp till 8 bonuspoäng

## PROTOTYP

- En prototyp är en körbar, men inte färdig, version av ditt program.
- Tanken är att du ska
  - komma igång med programmeringen
  - få in rutinen att bygga ut programmet successivt och testköra varje ny version.
- Prototyper kan ge upp till 8 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-UPPGIFTSLÖSNINGEN

- 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 kopierat
- 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 25 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 komplettera till högre betyg.
- Kompletteringen måste göras inom ett år från kursstart, dvs senast i januari 2015.



## SPEC - STRATEGI

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