

Programmeringsteknik

Föreläsning 13

- Allt om P-uppgiften

P-uppgiften: tider

- Fyra obligatoriska delmoment:
 - Spec - redovisas vecka 12
 - Prototyp - redovisas vecka 13
 - Granskning – en vecka innan du redovisar
 - Slutredovisning
 - senast 25 april för CL/Datorlingvistik
 - senast 10 maj för CSAMH

Välj P-uppgift

- P-uppgiften är ett **individuellt** moment.
- Uppgifterna är markerade så här:
 - Grunduppgift lätt/medel/svår, betyg E-A
- Varje grupp har en mapp med uppgifter att välja bland.
- Du måste välja ur din egen grupps mapp eller hitta på en egen (måste godkännas före spcredovisningen).

Specifikation

- Specen ska läggas in i Wuala och redovisas muntligt på grupptimmen vecka 12.
- Specen ska beskriva datastrukturer, funktioner, klasser (med attribut och metoder) och algoritm.
- Rita också (för hand) en minnesbild över datastrukturer i programmet.
- Specen kan ge upp till 8 bonuspoäng

Specen: obligatoriskt

Detta måste finnas med:

- Beskrivning av vad programmet ska göra
- Beskrivning av datastrukturer
- Uppdelning av problemet i funktioner
- Beskrivning av minst en klass
- Klassens attribut
- Klassens metoder

Specen: bonusgivande

- Bild av datastrukturen
- Alla funktioner, och alla metoder
 - har beskrivande kommentarer
 - har beskrivning av indata (parametrar)
 - har beskrivning av utdata (returvärden)
- Alla attribut har beskrivande kommentarer

Prototyp

- Prototypen ska redovisas i labbsal.
- Prototypen ska vara ett körbart Python-program – en första version av din uppgift.
- Tanken med prototypen är att du ska
 - komma igång med programmeringen
 - arbeta med att bygga ut programmet successivt och testköra varje ny version.

Prototyp: obligatoriskt

Detta är obligatoriskt:

- Prototypen ska vara upplagd i Wuala
- Prototypen ska gå att köra
- Prototypen ska innehålla stubbar* för klasser och funktioner enligt specen

* Stubbar har def-rad, kommentar men innehåller bara en pass- eller return-sats

Prototyp: bonusgivande

- Funktionsanropens inbördes ordning i main
- Kommentarer till alla funktioner/metoder
- Klassen/klasserna har
 - metoden `__init__`
 - metoden `__str__`
- Exempel på indatafil (i förekommande fall)
- Utskrift av en datastruktur (med värden)
- Testkörning som kommentar i slutet
- Beskrivning av nästa steg i arbetet

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](#).
- Syftet med granskningen är att
 - ditt eget program ska få en besiktning före redovisningen
 - du genom att kritiskt granska en annans program ska få en ökad förståelse för hur man ska (och inte ska) programmera.

Redovisning

- Boka tid för redovisning på webben.
- Ta med följande till redovisningen:
 - Uppgiftslydelsen
 - 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.

Krav vid redovisningen

- Programmet ska uppfylla uppgiftslydelsens krav.
- Du ska kunna redogöra för alla detaljer i programmet.
- Du ska kunna motivera designval.
- Programmet ska vara kommenterat upptill med författare och datum.
- Programmet ska vara användarvänligt.
- Programmet ska vara uppdelat i lagom långa funktioner, och klasser.

Fler krav

- Namn på variabler och funktioner ska vara vettiga.
- Nästan identiska kodstycken ska inte upprepas. Gör i stället generella funktioner.
- Inför konstanter istället för att skriva samma tal på flera stället.
- Varje variabel och funktion/metod ska vara kommenterad.
- För funktioner bör man också ange vad indata (parametrar) och utdata (returvärde) betyder.

Betyg

E = P-uppgift som lösts tillräckligt bra (max tre påpekanden).

D = Ett perfekt program, dvs inga anmärkningar i protokollet.

För betyg högre än D krävs att grunduppgiften redovisas före kursomgångens slut.

C = Kraven för D + en extrauppgift med betyg C (ofta hantering av felaktig inmatning).

B = Kraven för C + extrauppgift med betyg B (svårare algoritm).

A = Kraven för B + en extrauppgift med betyg A (grafik eller avancerad algoritm).

Observera att kraven på ett perfekt program gäller hela programmet, inklusive extrauppgift.

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.
- Konsultera Linda om du vill göra ett grafiskt gränssnitt istället för betyg B-uppgiften.

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 2013.