

Programmering og Problemløsning

Datalogisk Institut, Københavns Universitet

Arbejdsseddel 2 - individuel opgave

Jon Sparring

10. september - 18. september.
Afleveringsfrist: lørdag d. 18. september kl. 22:00.

Kom igang med F#

I sidste periode arbejdede vi med Scratch, som er et imperativt programmeringssprog. Nu skal vi i gang med F#, som vil være det sprog, vi vil benytte på resten af kurset. Vi starter med imperativ programmering, og I vil se, at nogle af de elementer, som I stiftede bekendtskab med i Scratch også bruges i F#. Som med \LaTeX skal vi bruge en teksteditor og kommandolinjen (terminalen) til at skrive og køre programmer. Der findes også hjælpeværktøjer kaldet Integrated Development Environments (IDE), som I vil stifte bekendtskab med senere på jeres uddannelse, og som kan gøre programmering nemmere, men her i starten skal I altså bruge kommandolinjen.

Emnerne for denne arbejdsseddel er:

- Kende forskel på en fortolker og en compiler, lave et fortolket og et oversat program,
- demonstrere evnen til at arbejde med strenge inkl. danske bogstaver, og
- kende forskellen på forskellige typer, og kunne konvertere heltal mellem decimal, binær, hexadecimal og oktal form.

Opgaverne er opdelt i øve- og afleveringsopgaver. I denne periode skal I arbejde individuelt med jeres afleveringsopgaver. Regler for gruppe- og individuelle afleveringsopgaver er beskrevet i ”Noter, links, software m.m.” → ”Generel information om opgaver”.

Øveopgaver (in English)

- 2ø0 Start an interactive F# session and type the following ending with a newline: `3.14+2.78;;`
Describe what F# did and if there was an error, find it and repeat.

2ø1 Repeat Exercise 2ø0, but this time, type the code in a text editor and save the result in a file with the suffix `.fsx`. Run through `fsharpi` from the console, and by first compiling it with `fsharpc` and executing the compiled file using `mono`. Consider whether the result was as expected and why.

2ø2 Write an expression which concatenates the strings `"hello"`, `" "`, `"world"` and run it in F#.

2ø3 Consider the F#-expression `"hello\nworld\n"`. Explain what the `"\n"` means, compute the expression using F# and discuss the result.

2ø4 Write an F#-expression for a string that contains the characters `"edb"` solely by using unicode escape codes.

2ø5 Write an F#-expression which extracts the 3. element and the substring from the 2. to the 4. element in the string `"abcdef"`.

2ø6 Write an F#-expression for a string which contains the character sequence `"\n"`, but where `"\n"` is not converted to a newline. How many different ways can this be done?

2ø7 Type the following expression in F# interactive mode, `3 + 1.0 ; ;` and explain the result. Consider whether you can improve the expression.

2ø8 Enter the integer 47_{10} on hexadecimal, octal, and floating-point form in F# and verify that all represents the same value.

2ø9 Consider the F#-expression `164uy+230uy`. Explain what `"uy"` means, compute the expression with `fsharpi`, and discuss the result.

2ø10 Using pen and paper:

- a) Write the integer 3_{10} on binary form by using the divide-by-2 algorithm.
- b) Write the integer 1001_2 on decimal form using the multiply-by-2 algorithm.
- c) Write the integer 47_{10} on hexadecimal and octal form.

Afløeringsopgaver (in English)

2i0 Describe the 3 ways an F# program can be run from the command line (terminal), and discuss the advantages and disadvantages of each method.

2i1 Using slicing, write an expression in F# which extracts the first and the second word from the string `"hello world"`. Enter the expression in an `.fsx` file and compile and run it. Does the program produce output? Explain why or why not.

2i2 Use pen and paper to complete the following table

| Decimal | Binary | Hexadecimal | Octal |
|---------|--------|-------------|-------|
| 10 | | | |
| | 10101 | | |
| | | 2f | |
| | | | 73 |

such that every row represents the same value written on 4 different forms. Include a demonstration of how you converted binary to decimal, decimal to binary, binary to hexadecimal, hexadecimal to binary, binary to octal, and octal to binary.

Krav til afleveringen

Afleveringen skal bestå af

- - en zip-fil, navngivet `fornavnEfternavnHoldXX-2i.zip`
- - en pdf-fil, navngivet `fornavnEfternavnHoldXX-2i.pdf`

Eksempel: Jon Sparring er på hold 1, så filerne hedder:

- `jonSparringHold01-2i.zip`
- `jonSparringHold01-2i.pdf`

Zip-filen skal indeholde to mapper: `src/` og `tex/`, samt filen `README.txt`. Mappen `src/` skal indeholde en enkelt fil navngivet `2i1.fsx`, der indeholder `fsharp`-koden til opgave 2i1. `2i1.fsx` skal kunne oversættes med `fsharpc`, og den oversatte fil skal kunne køres med `mono`. Mappen `tex/` skal indeholde `LATEX`-koden. Filen `README.txt` skal ganske kort beskrive, hvordan rapporten oversættes til pdf og koden oversættes og køres.

God fornøjelse.