MC346 2s2021

Horário

As aulas serão **sincronas** nos seguintes horarios:

3a: das 16h as 18h

5a: das 16h as 18h

Eu vou tentar limitar a duração das aulas a no maximo 90 minutos, ou seja muito provavelmente as aulas serão das 16 as 17 ou 17:30 horas. A gravação das aulas ficara disponível logo após a aula.

As aulas serão via google meet: https://meet.google.com/azk-tbag-gqo (https://meet.google.com/azk-tbag-gqo)

Inicio das aulas

dia 10/Agosto as 16 horas.

Descrição da disciplina

O objetivo da disciplina é apresentar ao aluno linguagens de programação que diferem de forma significativa das linguagens que ele já conhece, Python, C, e Java. Estas linguagens apresentam um conjunto de conceitos (também chamado de paradigmas) que vão expandir

as formas com que um programador pode pensar na solução para um problema.

A disciplina abordará as seguintes linguagens de programação, nesta ordem:

- Haskell (https://www.haskell.org/), particularmente a implementação GHC 8.X ou GHC 9.X para a parte de linguagens funcionais.
- Prolog, mais especificamente a implementação SWI-Prolog (http://www.swi-prolog.org), para a parte de linguagens lógicas.
- Alguns tópicos avançados em Python (versão 3.7 ou maior). Embora Python não seja por si só um exemplo de um novo paradigma de programação, ela nos permitirá discutir outros conceitos de programação em uma linguagem imperativa/tradicional. A disciplina assume que o aluno já sabe Python básico.
- Breves conceitos sobre algumas linguagens recentes como Go e Rust (linguagens que substituem C), Scala e Julia (linguagens no nivel de abstração de Python), e talvez Closure, e ML ou F# (versões menos estritas de linguagens funcionais).

Avaliação

Haverá 12 ou mais testes. Cada teste recebe as notas 0 se há mais de um erro ou um erro mais serio, 1 se há apenas um erro de menor impacto, e 2 se não há erros.

Os testes serão postados nas 3a ferias e o aluno tem 24 horas para submeter a sua resposta (provavelmente via Susy).

Haverá ainda 3 projetos a serem entregues provavelmente via Susy ou via email. Os projetos terão notas entre 0 e 1, proporcional ao número de testes corretamente executados. Os projetos poderão ser feitos em grupos de até 3 pessoas.

A nota final será a (soma das 8 maiores notas dos testes e as notas dos 3 projetos com peso 4) tudo dividido por 2.8

```
final = (8 melhores testes + 4*(soma dos 3 projetos))/2.8
```

Não há substitutiva para os testes e os projetos não poderão ser entregues atrasados.

Alunos que tiverem feito pelo menos 6 testes e tiverem uma nota final entre 2.5 e 4.9 poderão fazer o exame. O exame devera ser feito em 24 horas e submetido via email.

Datas importantes

(segundo email da coordenacao de graduação)

- Término das aulas 2s2020: 19/01/2021
- Último dia para para solicitação de desistência de matrícula em disciplinas: 07/10/2021
- Avaliação e Discussão dos Cursos: 27/10/2021
- Semana de estudos: 09 a 14/12/2021
- Exames finais do 2º período letivo de 2020 e Turmas Especiais I e II: 15 a 21/12/2021

Tarefas

As tarefas serão postadas aqui nas 3a feiras.

Aulas

- Apresentação: gravação (./aula0.mp4)
- Aula 1 (./aula1.html) dicotomias em linguagens de programação. gravação (./aula2.mp4)
- Aula 2 (./aula2.html) haskell basico. gravação (./aula3.mp4)
- Aula 3 (./aula3.html) Haskell basico 2 gravação (./aula4.mp4)
- Aula 4 (./aula4.html) Haskell tipos gravação (./aula5.mp4)

- Aula 5 (./aula5.html) Haskell funções de alto nivel gravação (./aula6.mp4)
- Aula 6 (./aula6.html) Haskell exemplos usando funcoes de alto nivel gravação (./aula7.mp4)
- Aula 7 (./aula7.html) Haskell - definindo tipos e lazy evaluation gravação (./aula8.mp4)
- Aula 8 (./aula8.html) Haskell Modulos gravação (./aula9.mp4)
- Aula 9 (./aula9.html) Haskell Containers/IO gravação (./aula10.mp4)
- Aula 10 (./aula10.html) OCaml e linguagens funcionais gravação (./aula11.mp4)
- Aula 11 (./aula11.html) Prolog - básico (unificação) gravação (./aula12.mp4)
- Aula 12 (./aula12.html) Prolog - básico 2 gravação (./aula13.mp4)
- Aula 13 (./aula13.html) Prolog números e listas gravação (./aula14.mp4)
- Aula 14 (./aula14.html) Prolog, if-then-else e cut gravação (./aula15.mp4)
- Aula 15 (./aula15.html) Prolog - predicados de alto nivel gravação (./aula16.mp4)
- Aula 16 (./aula16.html) Prolog - strings e I/O gravação (./aula17.mp4)
- Aula 17 (./aula17.html) Prolog final e CLP gravação (./aula18.mp4)
- Aula 18 (./aula18.html) Python - basico gravação (./aula19.mp4)
- Aula 19 (./aula19.html) Python exception e funcoes gravação (./aula20.mp4)
- Aula 20 - historia do python (./aula20-init.html) e aula 20 (./aula20.html) Python OO e prog funcional gravação (./aula21.mp4)
- Aula 21 (./aula21.html) Decorator e generators gravação (./aula22.mp4)
- Aula 22 (./aula22.html) Numpy basico gravação (./aula23.mp4)
- Aula 23 (./aula23.html) funcoes Numpy e cython gravação (./aula24.mp4)
- Aula 24 (./aula24.html) rust, go e scala gravação (./aula25.mp4)

Projetos

Os projetos serão listados aqui

Referencias

Haskell

Livro texto: Learn you a Haskell for greater good (http://learnyouahaskell.com/)

Uma lista de recursos (https://www.haskell.org/documentation/) para aprender Haskell

Prolog

- usaremos o SWI Prolog (http://www.swi-prolog.org/) versão 8.X
- Um livro texto de Prolog (http://www.learnprolognow.org/)

Python

- para instalar python (http://www.learnprolognow.org/) Não instale a versão 2.7. Usaremos a versão 3.7 ou maior. Uma alternativa para instalar o Python é usar o Anaconda Python (https://www.continuum.io/downloads) e talvez melhor ainda é instalar o Miniconda (https://conda.io/miniconda.html).
- Instale também o Numpy (https://www.numpy.org/) (já vem no Anaconda).
- Documentação do Python (http://docs.python.org/index.html)

Go

- site (https://golang.org/)

Rust

- site (https://www.rust-lang.org/)

Scala

- site (https://www.scala-lang.org/)

Julia

- site (https://julialang.org/)

Closure

- site (https://clojure.org/)

ML

- OCaml (https://ocaml.org/)
- F# (https://fsharp.org/)

Lisp

- um site (https://lisp-lang.org/)