MC346 Paradigmas de programação 2s2022

Horário

3a: das 16h as 18h CB16

5a: das 16h as 18h CB16

Inicio das aulas

dia 16/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, talvez não nesta ordem:

- Haskell, particularmente a implementação GHC 9.X para a parte de linguagens funcionais.
- 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.
- Linguagens da família Lisp: Lisp, Scheme e Closure
- Breves conceitos sobre algumas linguagens recentes como Go e Rust (linguagens que substituem C), Scala e Julia (linguagens no nível de abstração de Python), e talvez ML ou F# (versões menos estritas de linguagens funcionais).

Avaliação

Haverá pelo menos 10 tarefas. As tarefas terão de 3 dias a uma semana para serem feitas e devem ser submetidas via Google Classroom. Algumas tarefas serão individuais, algumas podem ser feitas em duplas – isto será indicado na tarefa.

A nota final será a média das 7 maiores notas das tarefas.

Não há substitutiva para as tarefas. Como a média só usa as 7 maiores notas, voce pode perder (e tirar 0) no mínimo 3 tarefas.

Alunos que tiverem feito pelo menos 5 tarefas e tiverem uma nota final entre 2.5 e 4.9 poderão fazer o exame.

O exame será dia 15/12 no horário de aula.

Tarefas

- Tarefa 0: Individual no classroom. Só testando se voce consegue acessar o classroom, e submeter um arquivo texto qualquer. (até 22/8 as 23:59)
- Tarefa 1: Individual ate 26/8 as 23:59 minha solucao
- Tarefa 2: Individual ate 30/8 as 23:59
- Tarefa 3: Individual ate 4/9 as 23:59

Aulas (plano)

- Aula 1 Dicotomias/escalas em linguagens de programação.
- Aula 2 Haskell básico.
- Aula 3 Haskell básico 2
- Aula 4 Haskell tipos
- Aula 5 Haskell funções de alto nível
- Aula 6 Haskell exemplos usando funções de alto nível
- Aula 7 Haskell - definindo tipos e lazy evaluation
- Aula 8 Haskell Modulos
- Aula 9 Haskell Containers/IO
- Aula 10 ML e outras linguagens funcionais
- Aula 11 Python - história e basico
- Aula 12 Python Exception, funções e OO
- Aula 13 Python modulos, prog funcional e debug
- Aula 14 Interators e generators
- Aula 15 Numpy
- Aula 16 Lisp 1
- Aula 17 Lisp 2
- Aula 18 Closure
- Aula 19 RUST
- Aula 20 GO
- Aula 21 Julia

Observações

Qualquer tentativa de fraude nas tarefas implicará em nota final 0 (zero) para todos os envolvidos, sem demais implicações. Exemplos de fraudes são:

- Compartilhar trechos de código de qualquer forma.
- Utilizar trechos de códigos da internet ou de outras fontes.
- Copiar ou comprar uma tarefa.

Versões Passadas

1 ano atras

2 anos atras

Referencias

Haskell

Livro texto: Learn you a Haskell for greater good

Uma lista de recursos para aprender Haskell

Python

- Não instale a versão 2.7. Usaremos a versão 3.7 ou maior. Uma alternativa para instalar o Python é usar o Anaconda Python e talvez melhor ainda é instalar o Miniconda.
- Instale também o Numpy (já vem no Anaconda).
- Documentação do Python

Closure

- site

ML

- OCaml
- F#

Scala

- site

Lisp

- um site

Go

- site

Rust

- site

Julia

- site