

MC336 – Paradigmas de Programação - 2s2011

Instrutor: [João Meidanis](#)

					I
	Notas (.html) (.xls)		Lista de discussão (só para alunos)		Email da Lista
Vídeos das aulas		Calendário Escolar 2o. semestre/2011			
NOVIDADES	2011-07-31	Site pronto			
	2011-07-30	Cronograma			
	2011-07-24	Site iniciado			
		Velhas novidades			
DATAS DAS PROVAS	Java: 08/09	Prolog: 20/10	Lisp: 01/12	Exame: 13/12	
Missão	Cronograma		Orientação a Objetos		
Visão	Atribuições do instrutor		Programação Lógica		
Avaliação	Atribuições dos alunos		Programação Funcional		
HORÁRIO E LOCAL	3ª feira 21:00-22:40 CB06		5ª feira 19:00-20:40 CB05		

Ementa:

Visão comparativa de paradigmas de programação. Programação funcional, lógica e orientada a objetos.

Missão

Dar aos alunos a oportunidade de conhecer e utilizar com desenvoltura outros paradigmas de programação diferentes daqueles aprendidos nas disciplinas introdutórias. Isto permitirá que tenham um arcabouço maior de recursos para resolver problemas em suas carreiras profissionais, o que é uma grande vantagem num mundo altamente mutante como o de hoje.

Visão

A disciplina está dividida em três partes, cada uma abordando um paradigma diferente de programação. Na primeira parte, será abordado o paradigma de **orientação a objetos**, exemplificado pela linguagem **Java**. Na segunda parte, será estudado o paradigma **declarativo**, através da linguagem **Prolog**. Por fim, na terceira parte será estudado o paradigma **funcional**, representado pela linguagem **Lisp**. Para melhor atingir os objetivos da disciplina, os recursos disponíveis na sala, como possibilidade de acesso a computador remoto, onde estarão instalados interpretadores ou compiladores das linguagens estudadas, serão usados para testar código ou observar estilos de programação.

A nota dos alunos será composta de duas partes. Uma das partes será constituída das provas individuais, onde os alunos deverão basicamente desenvolver trechos de código ou responder a questões teóricas a respeito dos

paradigmas e linguagens apresentados em aula. A outra parte corresponde a um projeto de programação onde os alunos desenvolverão jogadores para um certo jogo. Este desenvolvimento permitirá aos alunos avançarem em suas habilidades de programação e pensamento lógico.

Avaliação

Haverá dois tipos de avaliação: as provas escritas e os projetos de programação. As provas escritas valerão 70% da nota, enquanto que o projeto de programação entrará com os outros 30%.

As provas escritas terão 4 ou 5 questões, e duração de 1:40 horas. Haverá três provas escritas, uma para cada linguagem.

O projeto de programação consiste na implementação de um jogador para um certo jogo. Cada aluno individualmente escreverá um jogador, e os jogadores competirão entre si num campeonato, sendo que a colocação no campeonato decidirá a nota. Mais detalhes sobre o jogo e sobre o processo de avaliação dos jogadores podem ser encontrados numa página [específica](#).

O programa deverá ser entregue na data designada no cronograma. Programas entregues com atraso perderão pontos, a uma taxa de 0.0138889% por minuto de atraso, que equivale a uma taxa de 20% da nota por dia. Por exemplo, um programa entregue com 36 horas (2160 minutos) de atraso perderá 30% da nota. Um atraso de 5 minutos perderá apenas 0.07%, praticamente desprezível. Após 5 dias, não vale mais a pena entregar o programa.

Para escapar do exame, os estudantes deverão obter **nota de aproveitamento** (veja abaixo a definição) maior ou igual a 5.0 (cinco), e também uma nota maior ou igual a 4.0 (quatro) em **cada prova**.

Desta forma, a nota final de cada estudante será dada por:

Nome	Símbolo	Fórmula
Nota final	NF	$(NA + NE)/2$, se fez exame, ou NA caso contrário
Exame	NE	nota obtida no exame final (de 0 a 10)
Aproveitamento	NA	$0.7*PE + 0.3*PP$
Média das provas escritas	PE	$(PE1 + PE2 + PE3)/3$
Prova escrita 1	PE1	nota obtida na prova escrita 1 (de 0 a 10)
Prova escrita 2	PE2	nota obtida na prova escrita 2 (de 0 a 10)
Prova escrita 3	PE3	nota obtida na prova escrita 3 (de 0 a 10)
Nota do projeto	PP	nota obtida no projeto de programação (de 0 a 10).

Caso NF seja maior que 10, será rebaixada para 10.