UNIVERSIDADE FEDERAL DE SANTA CATARINA
C E N T R O T E C N O L Ó G I C O
Departamento de Informática e de Estatística

P L A N O D E E N S I N O

1. IDENTIFICAÇÃO DA DISCIPLINA: **Semestre**: 2001/2
Código: INE5612 **Nome**: Desenvolvimento de Sistemas Orientados a Objetos II
Horas/Aula: 72 **Teóricas**: 36 **Práticas**: 36
Código(s) do(s) pré-requisito(s): INE5609

2. OBJETIVOS:

2.1 - Gerais

Estudar conceitos e técnicas de engenharia de software pertinentes ao desenvolvimento de componentes de software.

2.2 - Específicos

Estudar técnicas específicas de desenvolvimento de componentes de software;
Exercitar o desenvolvimento de componentes de software.

3. PROCEDIMENTOS DIDÁTICOS:

(**AEX**=Aula expositiva; **LAB**=Aula de laboratório; **APR**=Aula prática; **OTR**=Outros)

T Ó P I C O S	Proc. Didático	Horas
1. Componentes de software	AEX	2
2. Metodologias de desenvolvimento	AEX	4
3. Estratégias de implementação	AEX	4
4. Estudos de casos (seminários)	AEX	20
5. Experimentos	APR	36
6. Discussões	AEX	6

4. AVALIAÇÃO DA APRENDIZAGEM:

Um seminário e um trabalho prático;
Recuperação através de uma única prova escrita sobre todo o conteúdo da disciplina.

5. BIBLIOGRAFIA:

Don Batory and Sean O'Malley, *The Design and Implementation of Hierarchical Software Systems with Reusable Components*. ACM Transactions on Software Engineering and Methodology, 1(4):355-398, 1992.

Ted J. Biggerstaff, *A Perspective of Generative Reuse*. Annals of Software Engineering, 5:169-226, 1998.

Special Section on Component-Based Enterprise Frameworks. Communications of the ACM, 43(10):24-66, 2000.

James O. Coplien, *Multi-Paradigm Design for C++*, Addison-Wesley, 1998.

Krysztof Czarnecki and Ulrich Eisenecker, *Generative Programming: Methods, Tools, and Applications*, Addison-Wesley, 2000.

Antônio A. Fröhlich, *Application-Oriented Operating Systems*, Sankt Augustin: GMD - Forschungszentrum Informationstechnik, 200 p., 2001.

William H. Harrison and Harold Ossher, *Subject-oriented Programming (a Critique of Pure Objects)*. in In Proceedings of the 8th Conference on Object-oriented Programming Systems, Languages and Applications, pages 411-428. Washington, U.S.A., 1993.

Ian M. Holland, *Specifying Reusable Components Using Contracts*. in Proceedings of the European Conference on Object-oriented Programming, pages 287-308. Springer, Utrecht, The Netherlands, 1992.

Ivar Jacobson, Grady Booch and James Rumbaugh, *The Unified Software Development Process*. Addison-Wesley, 1999.

Mehdi Jazayeri at al., *Generic Programming*. in Report of the Dagstuhl Seminar on Generic Programming, Schloß Dagstuhl, Germany, 1998.

Ralph E. Johnson, *Frameworks = (Components + Patterns)*. Communications of the ACM, 40(10):39-42, 1997.

Gregor Kiczales at al., *Aspect-Oriented Programming*. in Proceedings of the European Conference on Object-oriented Programming'97, pages 220-242, 1997.

Barbara Liskov and Stephen Zilles, *Programming with Abstract Data Types*. ACM SIGPLAN Notices, 9(4):50-59, 1974.

Mira Mezini and Karl Lieberherr, *Adaptive Plug-and-play Components for Evolutionary Software Developmen*. In Preceedings of the Conference on Object Oriented Programming Systems Languages and Aplications, pages 97-116, 1998.

David Lorge Parnas, *On the Design and Development of Program Families*. IEEE Transactions on Software Engineering, SE-2(1):1-9, 1976.

P. J. Plauger, *The Standard Template Library*. C/C++ Users Journal, 13(12):20-24, 1995.

Johannes Sametinger, *Software Engineering with Reusable Components*. Springer, 1997.

Yannis Smaragdakis and Don Batory, *Implementing Reusable Object-Oriented Components*. in Proceedings of the Fifth International Conference on Software Reuse, 1998.

Bjarne Stroustrup, *C++ Programming Language*. IEEE Software (special issue on Multiparadigm Languages and Environments), 3(1):71-72, 1986.

Bjarne Stroustrup, *The Design and Evolution of C++*. Addison-Wesley, 1994.

Bjarne Stroustrup, *The C++ Programming Language*. Addison-Wesley, 1997.

Clemens Szyperski and Rudi Vernik Establishing, *System-Wide Properties of Component-Based Systems: A Case for Tiered Component Frameworks*. in Proceedings of the Workshop on Compositional Software Architectures, 1998.

Michael VanHilst and David Notkin, *Using C++ Templates to Implement Role-Based Designs*. in Proceedings of the Second International Symposium on Object Technologies for Advanced Software, pages 22-37, 1996.

Michael VanHilst and David Notkin, *Decoupling Change from Design*. ACM SIGSOFT Software Engineering Notes, 21(6):58-69, 1996.

Todd L. Veldhuizen, *Using C++ Template Metaprograms*. C++ Report, 7(4):36-43, 1995.

Peter Wegner, *Classification in Object-oriented Systems*. ACM SIGPLAN Notices, 21(10):173-182, 1986.

David M. Weiss, *Software Synthesis: The FAST Process*. in Proceedings of the International Conference on Computing in High Energy Physics, 1995.

David M. Weiss and Chi Tau Robert Lai, *Software Product-line Engineering: A Family-Based Software Development Process*. Addison-Wesley, 1999.

Niklaus Wirth, *Program Development with Stepwise Refinement*. Communications of the ACM, 14(4):221-227, 1971.