

PMR2300 - Computação para Automação

Fabio Gagliardi Cozman, fgcozman@usp.br

PMR2300

Escola Politécnica da Universidade de São Paulo

- Responsáveis: Fabio G. Cozman, Newton Maruyama;
- Aulas teóricas às segundas-feiras, 10:10-12:00, e quartas-feiras, 10:10-11:00.
- Aulas práticas às segundas-feiras, 13:10-15:50, salas A4 e A6.
- A disciplina tem 5 aulas de laboratório de 3 créditos (perfazendo 15 créditos), e 15 aulas teóricas de 2 créditos e 15 aulas teóricas de 1 crédito (perfazendo 45 créditos);

Critério de Aprovação

- $M = \frac{5T+L}{6}$, onde $T = \frac{P1+2*P2}{3}$, e L é a nota de laboratório (exercícios-programa), desde que $L \geq 5$. Caso $L < 5$, $M = L$.

- Dar uma visão geral sobre ferramentas utilizadas para compor programas de grande porte de forma profissional.
- Ensinar:
 - Java
 - Orientação a objetos
 - Análise de complexidade e recursão
 - Algoritmos de ordenação
 - Estruturas de dados básicas

O desastre do Ariane 501

- Explosão do foguete Ariane 501:
“On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded.” (Comissão que investigou o acidente.)
- Principal causa:
“The failure of the Ariane 501 was caused by the complete loss of guidance and attitude information 37 seconds after start of the main engine ignition sequence (30 seconds after lift-off). This loss of information was due to specification and design errors in the software of the inertial reference system.”

Conclusões da comissão

- “The same requirement does not apply to Ariane 5, which has a different preparation sequence and it was maintained for commonality reasons, presumably based on the view that, unless proven necessary, it was not wise to make changes in software which worked well on Ariane 4.”
- “This means that critical software - in the sense that failure of the software puts the mission at risk — must be identified at a very detailed level, that exceptional behaviour must be confined, and that a reasonable back-up policy must take software failures into account.”

Em geral,

- Enfoque cada programa de forma modular e organizada.
- Documente suas ideias — documentação pode ser simples!
- Documente antes de programar, durante e depois de programar.
- Utilize bibliotecas sempre que possível.
- Programe pensando e verificando erros.

- Java: Ivan Mecenas, Java 2 - Fundamentos, Swing e JDBC, Alta Books, 2003;
- Teoria: Notas de Aula (disponível no site <http://sites.poli.usp.br/p/fabio.cozman/Didatico/pmr2300.html>);
- Suporte: M. T. Goodrich e R. Tamassia, Estruturas de Dados e Algoritmos em Java, Ed. Bookman, 2001.