

Critérios de correção dos trabalhos

Objetivos dos trabalhos

O objetivo da disciplina é aprender a ler, compreender, projetar e redigir programas modulares de boa qualidade, obedecendo a um conjunto de padrões. Este aprendizado é adquirido e demonstrado através da realização de uma série de trabalhos *interdependentes*.

O objetivo dos trabalhos não é escrever algum programa, mas, sim, é desenvolver programas *modulares de boa qualidade* e que *comprovadamente* satisfaçam massas de teste previamente estabelecidas. Também não é objetivo verificar se o aluno conhece todas as sutilezas da linguagem de programação, ou dos algoritmos empregados, no entanto é claro que se espera um domínio razoável da linguagem e dos algoritmos. Ou seja, independentemente da dificuldade, do esforço e do tempo gasto pelo aluno, o que vai ser examinado ao corrigir os trabalhos é a *qualidade* destes, independentemente do número de horas que o aluno possivelmente tenha levado para desenvolver os programas.

De maneira geral os trabalhos são bastante trabalhosos e sua realização deve ser iniciada imediatamente ao receber o enunciado. Os enunciados deixarão margens para dúvidas. Sempre consulte o instrutor para tirá-las. O objetivo disto é simular o “mundo real” encontrado ao desenvolver programas em empresas.

O que será corrigido é o que foi entregue. Ou seja, caso sejam entregues componentes obsoletos ou errados, ou sejam esquecidos componentes, o aluno perderá os pontos correspondentes. O objetivo deste critério é induzir os alunos a tomarem cuidado ao compor a versão final a ser entregue.

Entrega do trabalho

- Os trabalhos deverão ser entregues via e-mail. Caso a mensagem não satisfaça qualquer um dos itens a seguir: **-2 pontos**
 - O assunto da mensagem (*subject*) deve ser: ***IdDiscip-Trabnn-idGrupo*** em que:
 - *IdDiscip* é o código da disciplina, ex. INF1301
 - *nn* é o número do trabalho,
 - *idGrupo* é formado por *n* conjuntos de duas ou três letras, cada conjunto identificando um dos *n* membros do grupo.
 - Exemplo: INF1301-Trab1-JBOGESP
 - OBS.: Num espaço de 48 horas será enviada uma resposta a quem enviou o trabalho, confirmando se o trabalho foi recebido corretamente.
 - O texto da mensagem deve identificar cada um dos autores fornecendo os respectivos: letras identificadoras, número de matrícula, nome e endereço e-mail.
 - Todos os arquivos que compõem o trabalho devem estar "zipados" em um único arquivo anexado à mensagem enviada.
 - O nome do arquivo .ZIP deve ser ***IdDiscip-Trabnn-idGrupo.ZIP*** conforme descrito e exemplificado acima.
 - Para evitar a disseminação de vírus e outros tipos de mal-ware, alguns provedores não permitem a inclusão de arquivos .exe nos arquivos .zip. neste caso adicione o sufixo .txt. Exemplo, se o arquivo era **Trab01.exe**, rebatize-o para **Trab01.exe.txt**
 - A codificação do arquivo anexado ao e-mail deve ser MIME.
- Para cada dia útil de atraso (domingos e feriados não são dias úteis, porém sábados e dias enforcados são dias úteis) é descontado: **1 ponto**. Obs. O término de um dia é às 6 horas do dia a seguir. Por exemplo, se o trabalho era devido no dia 7 e for enviado no dia 8 antes das 6 horas, não perde ponto, após as 6 horas perde.

Composição do trabalho

- “Grupos” formados por um único ou por quatro ou mais alunos: **-2 pontos**.

- Procure formar o seu grupo já na primeira aula. Leve em conta a afinidade e a disponibilidade de tempo de cada participante. O objetivo é aprender a organizar e realizar trabalho em grupo. Os trabalhos são dimensionados para 2 alunos trabalhando em grupo.
- **Não será permitido formar grupo com alunos de outra turma.**
- **O aluno obrigatoriamente terá suas provas e trabalhos corrigidos apenas pelo professor da turma na qual está matriculado.**
- Conteúdo da mensagem com vírus resulta em nota **zero**
 - Cuidado com as máquinas dos laboratórios, pois, por mais controle que se exerça, colegas ingênuos ou pouco éticos freqüentemente infectam estas máquinas.
- São exigidos tipicamente os arquivos a seguir. O enunciado do trabalho pode exigir arquivos e documentos complementares.
 - Programa executável
 - Módulos implementação fonte
 - Módulos definição fonte
 - Arquivo RELATORid.TXT – um arquivo por membro do grupo, identificado por *id* registro de trabalho realizado
 - Arquivo LEIAME.TXT explicando com se utiliza o programa.
 - Arquivos contendo diretivas (*scripts*) de teste
 - Arquivos adicionais constantes do enunciado do trabalho
- Se o conteúdo do arquivo ZIP estiver incompleto **-2 pontos**

Execução básica

- Se o `leia.me` não corresponde ao comportamento do programa: **-2 pontos**
 - Se o programa não corresponde ao enunciado: **-8 pontos**.
 - Se o programa não corresponde a uma implementação modular: **-8 pontos**.
 - Se o programa for plágio de algum outro a nota do trabalho será **Zero**.
 - Se o programa `.exe` não existe ou não dá partida: **-8 pontos**
- OBS: Recomendamos especial cuidado com o teste final, assegurando que o programa opere em máquina sem a infra-estrutura necessária para o ambiente de desenvolvimento. Não serão abertas exceções. Caso o programa não rode em máquinas que não possuam o ambiente de desenvolvimento instalado, o grupo perderá os oito pontos.**
- Se o programa não completa a execução (entra em *loop*, trava, cancela a execução, voa, etc.) em todos os testes usados: **-6 pontos**
 - Se o programa não completa a execução em **um ou mais** dos casos de teste usados: **-4 pontos**. OBS. o teste será repetido para que se tenha certeza de não se tratar de uma falha fortuita do instrutor e na ficha de avaliação será descrito, em linhas gerais, como proceder para gerar a falha.
 - O programa não executa o *script* de teste desenvolvido pelos instrutores no momento da correção: **-3 pontos**.
 - Não foram incluídos *scripts* de teste produzidos pelo grupo: **-2 pontos**.
 - Os *scripts* de teste são superficiais ou mal organizados: **-1 ponto**.
 - O programa não utiliza o arcabouço (*framework*) de teste fornecido: **-3 pontos**. O objetivo é aprender a utilizar componentes fornecidos por terceiros.
 - Outros problemas encontrados: **-1 ponto por problema**

Código fonte e documentação

Estude os padrões contidos nos apêndices do livro!

- Se algum dos arquivos não identificar os autores: **-2 pontos**
- Se os módulos fonte não correspondem ao programa executável: **-8 pontos**
- Se outros arquivos não corresponderem ao programa implementado: **-4 pontos**
- O programa fonte não existe ou não foi escrito em C: a nota do trabalho será **zero**

- Comentários inexistentes ou fora dos padrões: **-1 ponto**
- Nomes dos elementos fora dos padrões **-2 pontos**
- Constantes fora dos padrões, ex.: programa utiliza literais e “números mágicos” ao invés de constantes simbólicas **-2 pontos**
- Módulo de definição ou módulo de implementação fora dos padrões, ex.: o módulo de implementação contém a declaração ou definição de uma ou mais variáveis globais externas; o módulo de definição não pode ser utilizado tanto para compilar o módulo servidor cuja interface define, nem para compilar os clientes desse módulo: **-2 pontos**
- Estilo do código fora dos padrões **-2 pontos**. OBS. caso você tenha um padrão de estilo *documentado (impresso)* e revisto por terceiros (ex. padrão adotado em alguma empresa) você poderá utilizá-lo, desde que entregue uma cópia do documento ao instrutor.
- Especificações de funções não existem **-2 pontos**
- Especificações de funções estão incompletas **-1 ponto**
- Especificações de funções são inconsistentes com a implementação do elemento especificado: **-2 pontos**
- Especificações de funções mal escritas (português incorreto, ambíguo, confuso): **-1 ponto**
- Programa mal organizado **-1 ponto**
- Código duplicado, mesmo que ligeiramente alterado: **-2 pontos**
- Interfaces entre módulos, classes e funções inutilmente complexas: **-1 ponto**
- Código longo sem conter pseudo-operações: **-1 ponto**
- Código usa funções que retornam condições de funcionamento e que não são verificadas pelo programa (ex.: não verifica se `malloc` ou `fopen` retornaram `NULL`): **-1 ponto**
- Usualmente o enunciado contém critérios complementares! Cada um deles identificará o número de pontos perdidos caso não seja satisfeito.

Observação

Evidentemente o total de pontos que um aluno pode perder segundo os critérios acima é maior do que 10. Entretanto, a nota de cada trabalho é independente das notas e dos pontos perdidos em outros trabalhos. Porém, como os trabalhos são interdependentes, um primeiro trabalho mal feito pode prejudicar todos os outros. No mínimo acarretará um esforço muito maior nos trabalhos subsequentes. Este esforço é necessário para corrigir os problemas identificados nos trabalhos anteriores.

Não deixem os trabalhos para o último dia! Eles dão muito trabalho!