

Trabalho I – Implementação de um Sistema Utilizando Conceitos de Programação Orientada a Objetos

Para o primeiro trabalho da disciplina de Programação Orientada a Objetos, os alunos deverão implementar um sistema utilizando a linguagem de programação Java. O sistema a ser desenvolvido deverá permitir a interação com o usuário através do console – utilizando o teclado – e ser capaz de manter (adicionar, alterar, consultar e remover) cadastro de dados referentes a uma problemática escolhida.

Não há restrição para o tema escolhido para o trabalho desde que os alunos consigam cumprir as exigências que já são propostas. Isto é, o trabalho poderá variar desde um sistema de informação tradicional para controle de biblioteca, escola ou loja, por exemplo, até aplicações mais interativas como um jogo de RPG ou Super Trunfo®, por exemplo. Os alunos podem optar também por utilizar um problema que já foi apresentado nos exercícios, estendendo e modificando os detalhes do problema. A criatividade e a pertinência dos sistemas desenvolvidos também serão um critério de avaliação.

A fim de utilizar os conceitos aprendidos até então na disciplina de Programação Orientada a Objetos, os alunos deverão adaptar a problemática escolhida de tal forma que os seguintes requisitos sejam satisfeitos:

- Criar pelo menos um tipo enumerável para ser utilizado no problema;
- Utilizar pelo menos uma composição de classes (i.e. pelo menos um atributo de uma classe deve ser um objeto de outra classe que você definiu);
- Utilizar pelo menos duas relações de herança que sejam válidas e úteis para o problema trabalhado. Não serão aceitas relações de herança sem sentido ou que não sejam úteis para a codificação. Seja criativo e em caso de dúvidas, procure e converse com o professor ou o monitor da disciplina;
- Se possível e aplicável, utilizar pelo menos uma classe abstrata;
- Os dados a serem mantidos pelo sistema deverão ser guardados e consultados em memória – utilize *arrays* de tamanho máximo especificados por uma constante.
- Definir pelo menos uma interface que define os comportamentos de persistência dos dados – ex: `addContact()`, `alterContact()`, `findContact()` – e fazer com que uma classe implemente esta interface, salvando os dados em memória.
- Criar pelo menos duas situações onde ocorra sobrecarga (*overloading*) e sobrescrita (*overriding*) de métodos;

Material a ser entregue: projeto Netbeans ou Eclipse contendo os arquivos de código fonte.

Restrições:

- Este trabalho poderá ser realizado no máximo em duplas
- Este trabalho deverá ser entregue no dia **07/05/2018** – até o início da aula
- Perguntas presenciais poderão ser realizadas para validar a autoria dos códigos