



Universidade Federal de Uberlândia
Curso de Graduação em Gestão da Informação
6º Trabalho de Estruturas de Dados I – Prof. Daniel A. Furtado
Trabalho Individual – Aplicação de Buscas

INTRUÇÕES GERAIS – LER ATENTAMENTE

- Esta atividade deve ser realizada individualmente;
- Este trabalho deve ser feito utilizando a linguagem C# ou Python. Trabalhos implementados utilizando outras linguagens serão anulados;
- **Todo o código deve ser colocado em um único arquivo**, pois apenas um arquivo (.cs ou .py) deve ser entregue. Não crie arquivos adicionais para as classes (o único arquivo a ser entregue deve conter todas as definições de classes, assim como o programa principal).
- **Não compactar** os arquivos no momento da entrega;
- Os recursos adequados das linguagens devem ser utilizados;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet, de trabalhos de colegas de classe ou de semestres anteriores serão anulados;
- As implementações não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato, etc.;
- O trabalho deve ser entregue até a data definida pelo professor em aula síncrona pelo Sistema de Aplicação de Testes (SAAT);
- Trabalhos enviados por e-mail ou por outros meios digitais (como o MS Teams) **não serão considerados** (veja instruções no final);

Este trabalho utiliza como base a implementação de lista ligada em *array* solicitada anteriormente no Trabalho 5. Crie um novo projeto em sua IDE de desenvolvimento aproveitando o código do referido trabalho.

1. Defina uma classe adicional de nome **Pessoa**, que contenha atributos públicos para armazenar as seguintes informações:
 - a. Código: inteiro
 - b. Nome: string
 - c. Idade: inteiro
2. Faça as alterações necessárias na classe referente à lista ligada em *array* para possibilitar o armazenamento de objetos do tipo **Pessoa**. Por exemplo, o *array* de itens que

armazenava inteiros deve ser redefinido para armazenar objetos da classe **Pessoa**. Alguns métodos devem ser adaptados para que possam receber, como parâmetro, objetos do tipo **Pessoa** (ao invés de um inteiro). Também será necessário adaptar as partes do código que fazem as comparações (buscas, algumas inserções e remoções). Utilize o campo “código” da classe **Pessoa** sempre que necessitar fazer comparações envolvendo os elementos da lista (por exemplo, utilizando construções como *items[i].Codigo != novaPessoa.Codigo*).

3. No programa principal, crie um “menu de opções” para que o usuário possa escolher uma ação da lista a seguir:
 - 1 – Cadastrar pessoa
 - 2 – Buscar pessoa no cadastro pelo código
 - 3 – Mostrar todas as pessoas cadastradas
 - 4 – Sair
- A **opção 1** deve permitir que o usuário informe novos dados de uma pessoa (Código, Nome e Idade). O programa deve coletar os dados do usuário, criar um novo objeto do tipo **Pessoa** utilizando os dados e inserir esse novo objeto ordenadamente (pelo código) na lista.
- A **opção 2** deve fazer uma busca na lista de pessoas utilizando como chave um código de busca que deve ser solicitado ao usuário. Caso o código seja encontrado, o programa deve mostrar os dados da pessoa sendo buscada. Caso contrário, deve mostrar uma mensagem informando que a pessoa não está cadastrada. A busca binária deve ser utilizada.
- A **opção 3** deve exibir os dados de todas as pessoas cadastradas.

Após executar uma opção escolhida pelo usuário, o programa deve voltar ao menu (exceto se for a opção 4 – Sair). Veja os exemplos a seguir para solicitar os dados ao usuário:

C#:

```
string nome = Console.ReadLine();  
int idade = Convert.ToInt32(Console.ReadLine());
```

Python:

```
nome = input("Informe o nome da pessoa: ")  
idade = int(input("Informe a idade: "))
```

Entrega

Um único arquivo (.cs ou .py) contendo todas as definições de classes e também o programa principal deve ser enviado pelo Sistema de Aplicação de Testes (SAAT) até a data limite

indicada pelo professor. Não envie arquivos compactados ou outros arquivos do Visual Studio que não sejam o arquivo de código principal.

Sobre Eventuais Plágios

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.