
PARADIGMAS DA PROGRAMAÇÃO IV

TRABALHO PRÁTICO Nº2

Entrega: 12 a 14 de Julho

Objectivos:

- Polimorfismo, Interfaces, Streams;
 - Reutilização;
-

AGENDA PROFISSIONAL

Considere as classes desenvolvidas para o 1º trabalho.

Modifique a classe **FichaPessoal** por forma a que agora possua apenas as seguintes informações:

- Nome e Morada;
- Telemóvel;
- Especialidades (lista, cf. “informática”, “design”, etc);
- Anos de conhecimento;
- Saldo (quanto deve ou quanto tem a receber).

Crie uma classe **FichaEmpresa** com a mesma estrutura da ficha anterior. Ambas as classes deverão ser *Cloneable* e *Serializable*, e deverão ainda ser ambas subclasses da classe abstracta **Ficha**, que apresenta a seguinte API:

- **Ficha(String nm, String mora, String tele);**
- **Ficha(String nm, String mora, String telem, Collection espec, int anos, double contas);**
- **String getNome/getMorada/getTelem**
- **Collection getEspec**
- **double getSaldo**
- **void mudaTele(String nvTele) throws NumTeleErradoException**
- **void mudaMora/mudaAnos(int inc)/mudaSaldo(int valor)**
- **void insereEspec(String nvEsp) throws EspecExistenteException**
- **int numEspec**
- **String toString()**

A classe **AgendaPro** implementa uma associação entre o nome de uma pessoa ou empresa e a respectiva ficha de informação. Esta classe deverá ser *Serializable* e deverá implementar a interface **Desfolhável**.

A interface **Desfolhável** é satisfeita por qualquer classe que implemente o seguinte comportamento:

- **void principio()** // posição inicial
- **void fim()** // posição final
- **void meio()** // posição média
- **void frente()** // avança 1 posição se possível
- **void trás()** // recua 1 posição se possível
- **int actual()** // indica a posição actual
- **Object getPagina** // dá a página actual

A classe **AgendaPro** implementa ainda as seguintes operações (na sua maioria já criadas no projecto anterior):

- Inserir uma nova página na Agenda;
- Remover uma página dado o nome (chave);
- Total de páginas;
- Total de páginas de empresas;
- Lista de todos os nomes na agenda;
- Modificar a ficha de nome dado (apagar a antiga e inserir a nova);
- A agenda deverá a qualquer momento poder ser gravada numa ObjectOutputStream e ainda permitir imprimir uma página dada num ficheiro de texto.

Execução do Projecto:

- Desenvolva e teste todas as classes usando o ambiente BlueJ;
- Crie em BlueJ a documentação das classes desenvolvidas (APIs);
- Crie um programa principal Java que realize o carregamento inicial de dados na agenda.

Entrega do Projecto:

- O projecto a entregar deverá ter um pequeno relatório, na capa do qual se identificam os elementos do grupo, a documentação gerada em BlueJ (APIs), e o projecto BlueJ;
- A pasta final de BlueJ será apresentada aquando da entrega do trabalho. Crie uma classe que lhe permita rapidamente apresentar uma agenda já construída, com, no mínimo 50 páginas, e que irá servir para testar os métodos e funcionalidades pedidas. Cada grupo terá 20 minutos para apresentar o trabalho.
- Situações de erro de execução ou de impossibilidade de execução serão de imediato canceladas, não havendo lugar a entregas posteriores. Idealmente os grupos deverão trazer os seus projectos em disquete ou “flash disk”.

pp4.bomTrabalho2());
