

UML

- 1) Quando é apropriado modelar utilizando-se de composição? Por exemplo, por que não utilizar a notação de composição da UML para mostrar que um cachorro é composto de altura, peso, cor e data de nascimento?
- 2) Um pão fatiado é constituído de fatias de pão. A associação entre pão e suas fatias corresponde a uma composição ou uma agregação?
- 3) Prepare um diagrama de classes para um editor de documentos gráficos que admita agrupamento. Suponha que um documento consista de várias páginas. Cada página contém objetos de desenho, incluindo texto, objetos geométricos e grupos. Um grupo é simplesmente um objeto de desenho, podendo incluir outros grupos. Um grupo precisa conter pelo menos dois objetos de desenho. Um objeto de desenho pode ser um membro direto de, no máximo, um grupo. Objetos geométricos incluem círculos, elipses, retângulos, linhas e quadrados.
- 4) Construa um diagrama de classes em UML para uma livraria online. O sistema deve atender os seguintes requisitos:
 - A livraria deve vender livros, mantidos em um catálogo, através de pedidos feitos pela Internet.
 - Os livros devem ser armazenados em um estoque mantido por diferentes fornecedores.
 - Todo usuário do sistema deve poder adicionar livros em um carrinho de compras *online*, antes do pagamento (*checkout*).
 - O usuário deve poder remover itens do carrinho de compras.
 - O usuário deve poder cancelar pedidos feitos antes de serem enviados.
 - O usuário deve poder pagar por boleto ou por cartão de crédito.
 - O usuário deve poder criar uma conta de cliente, de modo que o sistema registre os detalhes do usuário, incluindo nome, endereço, cartão de crédito e etc.
 - O sistema deve manter uma lista de contas em sua base de dados.
 - Quando um usuário logar, ele deve ter sua senha conferida com uma lista de senhas armazenadas no sistema.

- O usuário deve ser capaz de buscar por livros através de vários métodos de busca (por título, autor, palavras chave e categoria e ver detalhes do livro).
- 5) Um usuário tem acesso a múltiplos arquivos, cada um acessível por um tipo de permissão de acesso. Descreva em UML o relacionamento entre as classes correspondentes.
 - 6) Uma pessoa pode ter várias assinaturas de revista. Uma revista possui vários assinantes. Para cada assinatura é importante acompanhar a data e o valor de cada pagamento, além da data atual de expiração. Projete o diagrama de classes que descreva a assinatura de revistas.

Java Generics

- 1) Escreva um método genérico que ordene listas genéricas, que implementam a interface List do Java, de acordo com um critério de ordenação arbitrário. **Obs.: Não utilize o método sort da classe utilitária Collections. Utilize apenas as operações da interface.**

```

«java interface»
List<E>
+ boolean add(E e)
+ void add(int index, E element)
+ boolean addAll(Collection<? extends E> c)
+ boolean addAll(int index, Collection<? extends E> c)
+ void clear()
+ boolean contains(Object o)
+ boolean containsAll(Collection<?> c)
+ boolean equals(Object o)
+ E get(int index)
+ int hashCode()
+ int indexOf(Object o)
+ boolean isEmpty()
+ Iterator<E> iterator()
+ int lastIndexOf(Object o)
+ ListIterator<E> listIterator()
+ ListIterator<E> listIterator(int index)
+ boolean remove(Object o)
+ E remove(int index)
+ boolean removeAll(Collection<?> c)
+ boolean retainAll(Collection<?> c)
+ E set(int index, E element)
+ int size()
+ List<E> subList(int fromIndex, int toIndex)
+ Object[] toArray()
+ T[] toArray(T[] a)

```

- 2) Considere uma interface `Appendable` que inclui um método `append`. Além disso considere duas classes `MyString` e `MyList`, as quais implementam `Appendable`. Deseja-se que o sistema permita que um objeto de `MyString` seja concatenado com um objeto `MyString` e um objeto de `MyList` possa ser concatenado com outro objeto de `MyList`, mas que não seja admitido a concatenação de tipos diferentes. Observando a declaração abaixo diga se a interface proposta satisfaz os requisitos especificados quanto a tipagem. Em caso contrário, ofereça uma solução que corrija o problema.

```
interface Appendable{  
  
    Appendable append(Appendable a);  
  
}
```

- 3) Escreva um método genérico que conte o número de elementos em uma coleção que tenha uma propriedade específica (por exemplo, inteiros ímpares, números primos, palíndromos, etc).
- 4) Escreva um método genérico que troque duas posições diferentes em um array.
- 5) Implemente uma árvore binária de busca genérica com iteradores para percorrimento em ordem, pré-ordem, pós-ordem e largura.
- 6) Um Sistema de Informações Geográficas precisa gerenciar e manipular mapas de países, os quais são compostos por diferentes objetos geográficos. Considere como objetos geográficos possíveis os limites dos estados, municípios, as rodovias, as ferrovias e os rios. Escreva uma classe que descreva um mapa que pode conter os diferentes objetos geográficos mencionados. Implemente um método da classe `Mapa`, com assinatura `void draw(Canvas c)`, que exiba cada um dos elementos geográficos em uma superfície de desenho de tipo `Canvas`. Não é necessário especificar a implementação do modo de exibição de cada tipo específico de objeto geográfico.
- 7) A linguagem Lisp manipula estruturas simbólicas denominadas `SExpressions`. Uma `SExpression` é um átomo, isto é um dado considerado indivisível ou então é uma estrutura composta, formada pela sequência `"("`, uma `SExpression s1`, `","`, uma segunda `SExpression s2` e `")"`.

Tais expressões simbólicas podem ser interpretadas, armazenadas e manipuladas na forma de uma lista genérica (ou lista de listas), onde cada nó da lista contém duas referências, que podem apontar para um nó ou para um átomo. Exemplos de listas válidas em Lisp são:

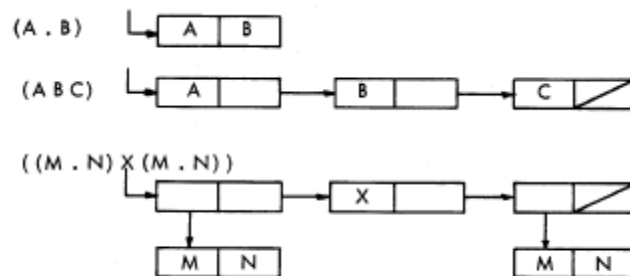


Figura obtida de [1]

Escreva uma classe genérica `LispList<T>` onde os átomos podem ser do tipo genérico `T` com as seguintes operações:

`cons` - retorna uma nova lista a partir da concatenação de um par de elementos que podem ser dois átomos, um átomo e uma lista ou duas listas existentes.

`car` - retorna o primeiro elemento da lista (cabeça).

`cdr` - retorna o restante da lista, excluindo o primeiro elemento (cauda).

Obs.: as operações `car` e `cdr` têm comportamento indefinido quando aplicadas sobre átomos. Lembre que as operações podem receber qualquer tipo de lista que seja de um subtipo de `T`.

[1] John McCarthy, Paul W. Abrahams, Daniel J. Edwards, Timothy P.Hart and Michael I. Levin. Lisp 1.5 Programmer's Manual – The Computation Center and Research Laboratory of Electronics – Massachusetts Institute of Technology. <http://www.softwarepreservation.org/projects/LISP/book/LISP%201.5%20Programmers%20Manual.pdf> em 03 de junho de 2013.

Coleções

- 1) Uma empresa de brinquedos armazena registros de todos os seus empregados contendo nome, CPF, data de ingresso e cargo. Como política da empresa, é comum a doação de certos benefícios aos empregados e uso dos nomes de alguns dos empregados nos seus produtos.

Considere as quatro interfaces principais da API Collections do Java, Set, List, Queue, and Map. Especifique quais das quatro interfaces é mais apropriada para armazenar os dados necessários para a execução de cada uma das atividades que deve ser realizada pelo sistema da empresa e explique a razão de sua escolha. As atividades são:

- Mensalmente um empregado deve ser escolhido aleatoriamente para receber um brinquedo grátis.
 - Cada novo produto lançado pela fábrica deve ser nomeado a partir do primeiro nome de um empregado. Na escolha, não devem ser consideradas duplicatas de nomes.
 - Os brinquedos especiais devem ser batizados com os nomes mais populares da empresa.
 - A empresa adquiriu ingressos para partidas de futebol que serão distribuídos aos empregados. A empresa mantém uma ordem de espera para distribuição dos ingressos aos empregados, ordem que depende do momento da solicitação do ingresso por cada empregado.
- 2) Implemente uma lista circular como uma especialização (customização) de `AbstractList<E>`
 - 3) Implemente o algoritmo de ordenação HeapSort que atue sobre uma coleção e que use um critério de ordenação que possa ser configurável para os elementos da coleção.
 - 4) Estude como é implementado o método `toArray()` da classe `ArrayList()`
 - 5) Considere um conjunto documentos na forma de arquivos texto. Escreva um programa que receba um conjunto de documentos e determine o grau de similaridade entre cada um deles. Cada documento deve disponibilizar um modo de medir sua similaridade quando confrontado a outro de mesma natureza. Cada documento é caracterizado por um perfil (*profile*) e os perfis devem ser capazes de

distinguir documentos diferentes. Existem inúmeras formas de se definir perfis. Uma possibilidade é usar a frequência dos elementos no documento. Um exemplo de perfil é construído com base no conceito de frequência de *k-grams*. Um *k-gram* é uma substring de tamanho *k* na string que compõe o documento. Para construir um histograma de *k-grams* é necessário associar um número a uma string (*hashing*). Perfis podem ser comparados através de uma métrica que meça a distância entre os vetores unitários associados ao histograma sendo duas estratégias bem simples as baseadas na métrica euclidiana e no produto escalar. Análise como sua solução responderia em face as seguintes mudanças:

- a) for preciso trabalhar com outros Documentos;
- b) for permitido usar diferentes formas de descrever os documentos ;
- c) for preciso usar métricas diferentes para comparar as similaridades entre os perfis;
- d) for permitida diferentes visualizações dos resultados na interface gráfica
- e) as configurações puderem ser feitas dinamicamente em tempo de execução

Princípios de Projeto O.O.

- 1) Analise as funções amigáveis (friends) de C++ do ponto de vista de congneridade.
- 2) Considere as duas linhas de código abaixo. Cite os possíveis tipos de congneridade que podem ocorrer exemplificando:

```
int i;  
i = 7;
```

- 3) Considere um jogo do tipo *Text Adventure* onde você pode navegar por um mapa usando comandos norte, sul, leste e oeste. Suponha que no programa as direções associadas aos comandos forma armazenadas na forma de atributos de uma classe com os seguintes valores

```
norte = 0; leste = 1; sul = 2; oeste = 3;
```

Que tipo de congneridade existe nesta situação.

- 4) Uma biblioteca de classes que você comprou de um distribuidor de classes gerais provavelmente conterá somente classes de base. Por que você acha que isso ocorre?
- 5) Defina os diferentes tipos de domínios em um sistema.
- 6) Porque classes não devem pertencer a mais de um domínio?
- 7) Determine a dimensionalidade do espaço de estados de um retângulo supondo que o mesmo seja capaz de girar, transladar e sofre escalas na altura e largura.
- 8) Defina o conceito de invariante de classe. Cite um exemplo.