

Laboratório de Engenharia de Software

Exercícios – Padrões de Projeto

1. O **padrão MVC** (*Model-View-Controller*) é um padrão de projeto mais complexo que é empregado principalmente em interfaces gráficas. Em anexo há um exemplo em Java de sua aplicação no arquivo MVC.zip (de <http://www.austintek.com/mvc/>). Pede-se:
 - (a) Desenhar um diagrama de classes UML correspondente ao exemplo fornecido.
 - (b) A partir deste diagrama, identificar padrões de software conhecidos que foram empregados em sua construção.
2. Elaborar, com o auxílio do **padrão Visitor**, um programa Java que, a partir de um nome de pasta fornecida, exiba na tela os nomes de arquivos e pastas que casem com uma expressão regular fornecida pelo usuário. Depois, elaborar um diagrama de classes UML equivalente ao programa desenvolvido.

Dica: para manipular arquivos ou diretórios do sistema operacional, utilize a classe `File` de Java e processe a árvore recursivamente.
3. Elabore um programa em Java que utilize o **padrão Strategy** para definir uma coleção parametrizada (por exemplo, um vetor) de modo que se possa utilizar diversos métodos de ordenação sobre essa mesma coleção (o usuário define). Exemplo:

```
MyVector<Integer> v = new MyVector<>(10);  
v.sort(new QuickSort<Integer>()); //Ordena com quicksort  
//...  
v.sort(new BubbleSort<Integer>()); //Ordena com bubblesort
```

4. Considere o programa Java a seguir:

```
import java.util.*;

public class IteratorDemo {
    public static void main(String args[]) {
        // Criar o array
        ArrayList<String> al = new ArrayList<>();
        // Adicionar elementos ao array
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add("D");
        al.add("F");

        // Exibir valores
        System.out.print("Valores de al: ");
        Iterator<String> itr = al.iterator();
        while (itr.hasNext()) {
            String element = itr.next();
            System.out.print(element + " ");
        }
        System.out.println();
    }
}
```

Pede-se: pesquise o **padrão Factory Method** e então tente identificar seu uso nas classes de coleções e de iteradores da biblioteca do Java, justificando sua resposta com um diagrama UML do padrão e como ele foi usado para resolver o problema de iteração sobre uma coleção.

5. Assuma que se tem um arquivo-texto contendo números de cartões de crédito. Cada linha deste arquivo contém os seguintes dados, separados por ponto e vírgula: o número do cartão, a data de expiração e o nome do titular do cartão. Elaborar um programa Java que utilize os seguintes nomes de classes a seguir:

- CreditCard;
- VisaCC, MasterCC, AmExCC, todas subclasses de CreditCard.

Futuramente, novas classes de cartão poderão ser adicionadas. Assim, pede-se

- (a) Implementar um programa Java que leia um arquivo-texto com diversos números de cartões. O programa deverá ter as classes descritas anteriormente, além de outras que se julgar necessárias para implementar o programa.
- (b) O programa deverá ler cada informação de cartão e então identificar que cartão é este, criando um objeto da classe correspondente na memória e o armazenando em uma estrutura de dados conveniente na memória. Depois, o programa deverá listar na tela os cartões identificados, agrupados por tipo (Amex, Visa etc). Tentar identificar e aplicar um ou mais padrões de projeto para serem adaptados a este projeto – não esquecer de escrever e justificar os nomes dos padrões na documentação a ser enviada.

NOTA: não é necessário utilizar o algoritmo exato para identificar os cartões (ISO 2894/ANSI 4.13). Utilize a **tabela** a seguir:

MasterCard	O primeiro dígito é 5 e o segundo dígito está no intervalo [1..5]. O comprimento é de 16 dígitos.
Visa	O primeiro dígito é 4. O comprimento é 13 ou 16 dígitos.
AmericanExpress	O primeiro dígito é 3 e o segundo dígito é 4 ou 7. O comprimento é de 15 dígitos.
Discover	Os primeiros quatro dígitos são 6011. O comprimento é 16 dígitos.

6. Pesquisar como o padrão **Decorator** poderia ser utilizado para resolver o problema da impressão de tickets comercializados por uma empresa de promoção, sabendo que:
- O ticket sempre possui um texto básico que deverá aparecer quando for impresso;
 - A empresa precisa adicionar, de modo flexível, diferentes cabeçalhos e/ou rodapés aos tickets comercializados, por imposição de seus clientes.

Propor um programa Java que simule esta situação e implemente o padrão **Decorator** para este caso.