

## Execução

- Grupos: exercício deve ser resolvido individualmente ou em duplas.
- Data limite de entrega: 15/04, antes do horário de início da aula, ou seja, até as 17:35h.
- Formato da entrega: via moodle, em arquivo “zipado” contendo o projeto implementado. O projeto deve incluir uma classe de teste que demonstre a solução do problema apresentado.

## Problema

A política básica de preços de “passaportes” de acesso a um grande parque de diversões é bastante simples: o valor é de US\$ 100,00 por dia, mas se o cliente comprar um passaporte válido por mais dias (até um máximo de 7) os dias subsequentes sofrem um desconto incremental de 20% ao dia (desconto de 20% sobre o valor do dia anterior). A interface Passaporte define este conceito e é implementado pela classe PassaporteBasico (códigos Java abaixo).

O parque, entretanto, com frequência faz convênios com diferentes empresas e inventa novas promoções. Todas as promoções são cumulativas, ou seja, se uma pessoa se enquadra em mais de uma situação pode acumular as vantagens. Atualmente estão em vigor as seguintes promoções:

- a) Estudantes universitários têm desconto adicional de 10% a partir do 2º dia na compra de passaportes para pelo menos 5 dias.
- b) Quem paga com cartão de crédito da bandeira “PagueBem” tem um desconto adicional de 2% sobre o valor total do passaporte.
- c) Pessoas com mais de 65 anos tem direito a um desconto adicional de 15% a partir do 3º dia na compra de passaportes para pelo menos 4 dias.
- d) Segurados da empresa “SeguroGarantido” tem desconto adicional de 2% sobre o valor total do passaporte na compra de passaportes para 7 dias.

~~O sistema deve manter uma coleção com todos os passaportes emitidos e implementar um contador de dias ainda válidos de cada um, que será utilizada como uma “cache global” em memória (só existe uma única instância dela, inicializada todos os dias antes do parque abrir) de forma a otimizar o acesso aos dados pelas catracas de entrada no parque. Um sistema de notificações faz com que esta “cache” seja atualizada cada vez que um novo passaporte é emitido. (esta parte está fora do escopo deste exercício)~~

Código inicial de apoio:

```
public interface Passaporte {
    String getNomeCliente();
    int getNroDias();
    double valorBasico();
    double valorDoDia(int nrodia);
    double valorTotal();
    GregorianCalendar dataInicial();
}

public class PassaporteBasico implements Passaporte{
    private String nomeCliente;
    private int nroDias;
    private double valorBasico;
    private GregorianCalendar dataInicial;

    public PassaporteBasico(String umCliente,int umNroDias, double
umValorBasico, int dia, int mês, int ano){
        nomeCliente = umCliente;
        nroDias = umNroDias;
        valorBasico = umValorBasico;
    }
}
```

```

        dataInicial = new GregorianCalendar(ano,mes,dia);
    }

    public String getNomeCliente() { return nomeCliente; }

    public int getNroDias() { return nroDias; }

    public double valorBasico() { return valorBasico; }

    public GregorianCalendar dataInicial() { return dataInicial.clone(); }

    public double valorDoDia(int nrodia) {
        double valorDia = valorBasico();
        for(int i=0; i<nrodia-1; i++){
            valorDia = valorDia - (valorDia*0.2);
        }
        return valorDia;
    }

    public double valorTotal() {
        double valorTotal = 0.0;
        for(int d=1;d<=getNroDias();d++){
            valorTotal += valorDoDia(d);
        }
        return valorTotal;
    }
}

```

## Implementação

- Conforme discutido em aula, este problema pode ser resolvido utilizando dois padrões de projeto:
  - “Factory method” para a instanciação dos passaportes, a partir dos dados básicos do passaporte (nome do cliente e número de dias) e quais os descontos aplicáveis (se for o caso).
  - “Decorator” para implementar o controle dos descontos. Lembre-se que os descontos são cumulativos.
- Trabalho alternativo – implementar uma solução para o problema proposto sem o uso dos padrões de projeto. Na minha opinião uma péssima ideia...