Lista de Exercício

1. Escreva um programa que carregue dois valores A e B pelo teclado e imprima todos os números ímpares entre A e B.

2. Escreva um programa que leia o nome e salário atual de um funcionário. O programa deve calcularseu novo salário (segundo a tabela abaixo) e mostrar o nome, o salário atual e o salário reajustado do funcionário:

Faixa salarial		**Acréscimo**
acima de	**até**	
--	150	25%
150	300	20%
300	600	15%
600	--	10%

- repita o processo acima até que seja digitado FIM no lugar do nome do funcionário;
- mostrar ao final do programa a soma dos salários atuais, a soma dos salários reajustados e adiferença entre eles.

3. Os números de Fibonacci formam uma sequência em que cada número é igual à soma dos dois anteriores. Os dois primeiros números são, por definição igual a 1, segundo o exemplo abaixo:

   Ex: 1 1 2 3 5 8 13 ...

   Escreva um programa que carregue um carregue um número inteiro pelo teclado e indique se ele faz parte da sequência de Fibonacci.

4. Escreva uma função que receba como parâmetro um número inteiro relativo a um mês do ano e retorne uma string com o nome deste mês por extenso. Resolva o problema de suas maneiras:

   - sem um vetor, através de uma estrutura switch/case;
   - com um vetor.

5. Escreva um programa que carregue um número inteiro e indique se ele é um número primo, para isto deve ser usado um módulo que recebe como parâmetro o número e retorna verdadeiro se ele for primo e falso caso contrário. Carregue um valor inteiro N pelo teclado e imprima os N primeiros números primos.

6. Crie uma classe a partir do desenho do diagrama da UML (Diagrama de Classes). Crie três contas e teste suas funcionalidades (atribuir valores aos atributos e executar suas operações).

Classe Conta
- numero: string - nome: string - saldo: int - limite: int
+ Conta( ) + Conta(anumero, atitular, asaldo, alimite) +Deposita (valor) +Saca(valor) +Extrato( )

7. Crie uma classe Funcionario com os seguintes atributos: código, nome, salário. Crie também um método para mostrar os dados. Digite os atributos e apresente-os.
8. Crie uma classe Produto com os seguintes atributos: código, descrição, preço. Crie também um método para mostrar os dados. Digite os atributos e apresente-os. **Ao final apresente o valor total obtido pela soma dos preços**.
9. Complemente a classe Conta com uma função, para fazer transferência entre contas. Para isso, na passagem de parâmetro, utilize um objeto do tipo conta.

Classe Conta
- numero: string - nome: string - saldo: int - limite: int
+ Conta( ) + Conta(anumero, atitular, asaldo, alimite) +Deposita (valor) +Saca(valor) +Extrato( )

10. Exercício

Criar uma classe aluno com os seguintes atributos: nome, ra, nota1, nota2.

Criar 2 tipos de construtores.

Criar os gets e sets para todos os atributos.

Criar um método que calcule a média das notas.

Criar 3 alunos e instancie-os de maneiras diferentes.

Para cada aluno exibir a média das notas.

11. Exercício

Criar uma classe retangulo com os seguintes atributos: bas e alt.

Criar 2 tipos de construtores.

Criar os gets e sets para todos os atributos.

Criar um método que calcule a área do retangulo.

Criar um método que calcule o perímetro do retangulo.

Criar 3 retangulos e instancie-os de maneiras diferentes.

Para cada retangulo exibir suas medidas, a área e o perímetro.

12. Exercício

Criar uma classe pessoa com os seguintes atributos: nome, idade e nacionalidade.

Criar 2 tipos de construtores.

Criar os gets e sets para todos os atributos.

Criar um método para verificar se a pessoa é maior de idade (idade =>18) ou se é menor de idade.

Criar um método que calcular quantas horas a pessoa já viveu.

Criar 3 pessoas e instancie-as de maneiras diferentes.

Para cada pessoa se é maior de idade e quantas horas já viveu.

13. Exercício

Criar uma classe funcionário com os seguintes atributos: nome, salario, cargo.

Criar 2 tipos de construtores.

Criar os gets e sets para todos os atributos.

Criar um método que calcule um reajuste salarial de 15% se o funcionário tem salario igual ou maior que 2000 reais, caso contrário o reajuste será de 20%..

Criar 3 funcionário e instancie-os de maneiras diferentes.

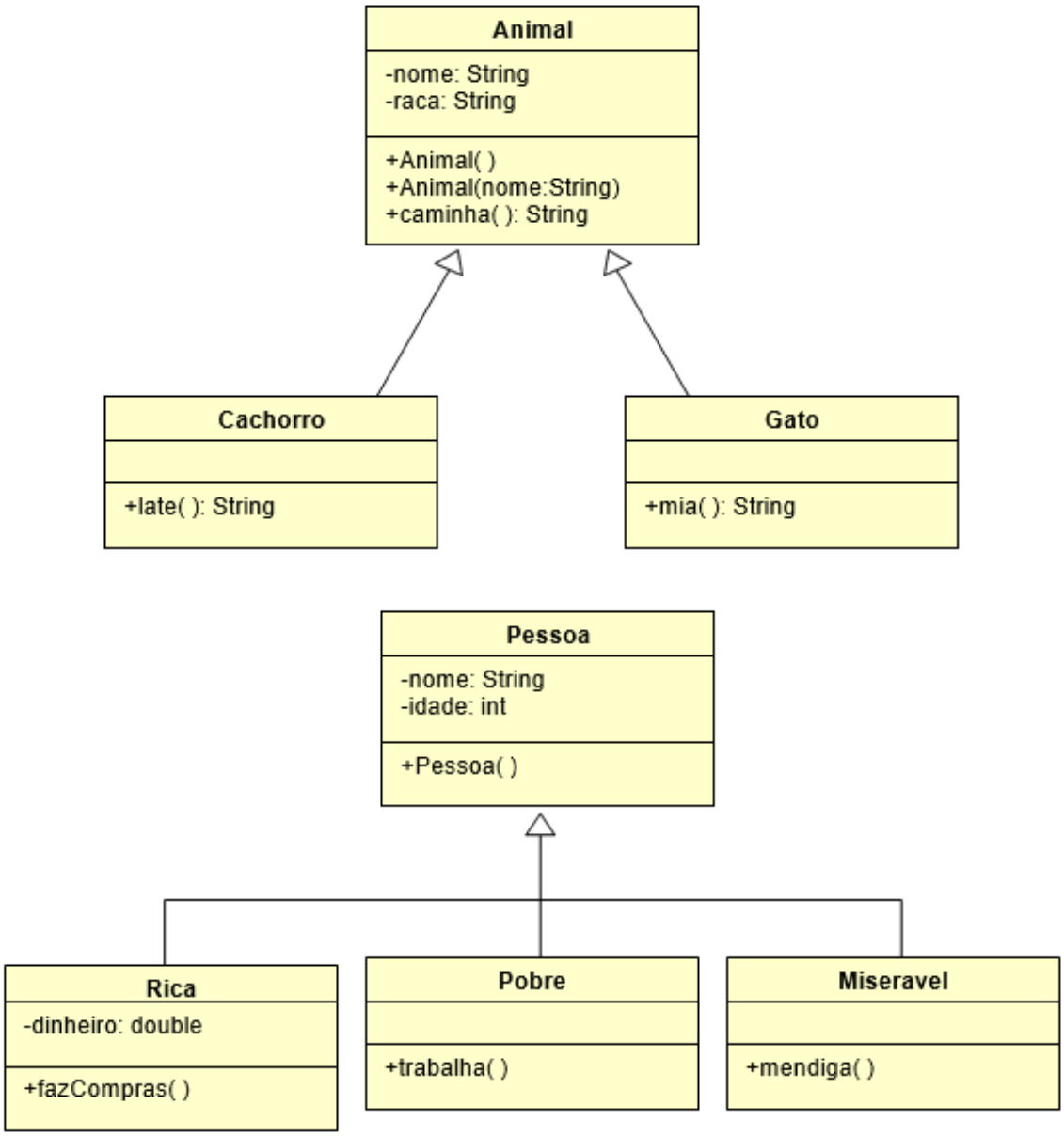
Para cada funcionário exibir os valores dos seus atributos, e o valor do salario reajustado.

Exercício 14: Implemente a classe Funcionario e a classe Gerente.

a. crie a classe Assistente, que também é um funcionário, e que possui um número dematrícula (faça o método GET). Sobrescreva o método exibeDados().

b. sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes Tecnico e Administrativo.

Exercício 15: Implemente os diagramas de classe abaixo:



Exercício 16: Crie uma classe chamada Ingresso que possui um valor em reais e um método imprimeValor().

a. crie uma classe VIP, que herda Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído).

b. crie uma classe Normal, que herda Ingresso e possui um método que imprime: "Ingresso Normal".

c. crie uma classe CamaroteInferior (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe CamaroteSuperior, que é mais cara (possui valor adicional). Esta última possui um método para retornar o valor do ingresso. Ambas as classes herdam a classe VIP.

Exercício 17: Crie a classe Imovel, que possui um endereço e um preço.

a. crie uma classe Novo, que herda Imovel e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.

b. crie uma classe Velho, que herda Imovel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

Exercício 18: Crie uma classe de Teste com o método *main*. Neste método:

a. crie um assistente administrativo e um técnico. Imprima o número de matrícula e onome de cada um deles.

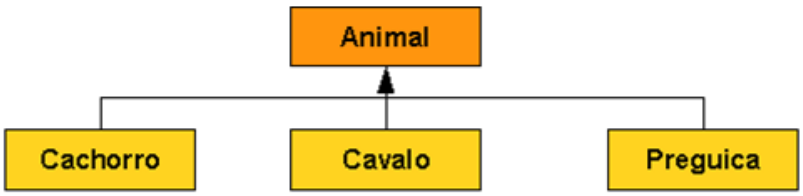
b. crie um animal do tipo cachorro e faça-o latir. Crie um gato e faça-o miar. Faça osdois animais caminharem.

c. teste (como quiser) as classes Rica, Pobre e Miseravel.

d. crie um ingresso. Peça para o usuário digitar 1 para normal e 2 para VIP.Conforme a escolha do usuário, diga se o ingresso é do tipo normal ou VIP. Se forVIP, peça para ele digitar 1 para camarote superior e 2 para camarote inferior. Conforme a escolha do usuário, diga se que o VIP é camarote superior ou inferior.Imprima o valor do ingresso.

e. crie um imóvel. Peça para o usuário digitar 1 para novo e 2 para velho. Conforme adefinição do usuário, imprima o valor final do imóvel.

19. Crie uma hierarquia de classes conforme abaixo com os seguintes atributos e comportamentos (observe a tabela), utilize os seus conhecimentos e distribua as características de forma que tudo o que for comum a todos os animais fique na classe Animal:



Cachorro	Cavalo	Preguica
Possui Nome	Possui Nome	Possui Nome
Possui Idade	Possui Idade	Possui Idade
Deve emitir som	Deve emitir som	Deve emitir som
Deve correr	Deve correr	Deve subir em árvores

20. Implemente um programa que crie os 3 tipos de animais definidos no exercício anterior e invoque o método que emite o som de cada um de forma polimórfica, isto é, independente do tipo de animal.

21. Implemente uma classe Veterinario que contenha um método examinar() cujo parâmetro de entrada é um Animal, quando o animal for examinado ele deve emitir um som, passe os 3 animais com parâmetro.

22. Crie uma classe Zoologico, com 10 jaulas (utilize um array) coloque em cada jaula um animal diferente, percorra cada jaula e emita o som e, se o tipo de animal possuir o comportamento, faça-o correr.

23. Resolva a seguinte situação utilizando os conceitos aprendidos. Uma empresa quer manter o registro da vida acadêmica de todos os funcionários, o modelo deve contemplar o registro das seguintes informações, de forma incremental:
    a. Para o funcionário que não estudou, apenas o nome e o código funcional;
    b. Para o funcionário que concluiu o ensino básico, a escola;
    c. Para o funcionário que concluiu o ensino médio, a escola;
    d. Para o funcionário que concluiu a graduação, a Universidade;

24. Estenda o modelo implementado no exercício anterior de forma que todo funcionário possua uma renda básica de R$ 1000,00 e:

a. Com a conclusão do ensino básico a renda total é renda básica acrescentada em 10%;
b. Com a conclusão do ensino médio a renda total é a renda do nível anterior acrescentada em 50%;
c. Com a conclusão da graduação a renda total é a renda do nível anterior acrescentada em 100%;
d. Todos os cálculos são efetuados sempre sobre a última renda obtida.

25. Crie um programa que simule uma empresa com 10 funcionários, utilize um array, sendo que a escolaridade dos funcionários é distribuída da seguinte forma: 40% ensino básico, 40% ensino médio e 20% nível superior. Calcule os custos da empresa com salários totais e por nível de escolaridade, utilize a classe funcionário desenvolvida no exercício anterior.

26. Faça uma hierarquia de Comissões, crie as comissões de Gerente, Vendedor e Supervisor. Cada uma das comissões fornece um adicional ao salário conforme abaixo:

1. Gerente: R$1500,00
2. Supervisor: R$600,00
3. Vendedor: R$250,00

27. Adicione a classe funcionário um atributo referente as comissões desenvolvidas no exercício anterior. Corrija o método renda total de forma que ele some ao valor da renda calculada o adicional da comissão do funcionário.

28. Refaça o exercício 20 considerando que 10% dos funcionários são Gerentes, 20% são supervisores e 70% são vendedores.

29. Sobreescreva o método toString de forma que ele imprima o nome do funcionário, a comissão e o salário total. Imprima todos os funcionários da empresa criada no exercício 20.

30. Exercício sobre Lista:

    Faça um programa que crie uma lista L, e contenha operações para:
    - Inserir um elemento em qualquer posição da lista;
    - Exibir os elementos da lista;
    - Remover um determinado elemento da lista;
    - Verificar a existência de um determinado elemento da lista, caso exista fornecer sua posição de armazenamento;
    - Alterar o valor de uma determinada posição da lista;
    - Exibir o tamanho da lista;
    - Esvaziar a lista.

31. Exercício sobre lista:
    Dada uma lista L1 de valores inteiros, que possam estar desordenadas e contendo elementos repetidos:

– Faça uma cópia da lista L1 em uma outra lista L2;
– Faça uma cópia da Lista L1 em L2, eliminando elementos repetidos;
– Inverta L1 colocando o resultado em L2;
– Gere uma lista L2 onde cada registro contém dois campos de informação: valor contém um elemento de L1, e qde contém quantas vezes este elemento apareceu em L1
– Elimine de L1 todas as ocorrências de um elemento dado.
– Forneça os elementos que aparecem o maior e o menor número de vezes (forneça os elementos e o número de vezes correspondente)

32. Faça uma função para inserir elementos em uma lista de forma ordenada (crescente ou decrescente), ou seja, a cada inserção o elemento deve ser inserido na posição correta de forma que a lista permaneça ordenada. Elabore o programa para testar essa função.

33. Defina a estrutura de dados para representar uma lista de alunos cujo nó tem os seguintes campos: Matrícula, nome e idade. Crie uma função para inserir os dados de um aluno na lista, onde matrícula é a chave.

34. Escreva uma classe cujos objetos representam alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: matrícula, nome, 2 notas de prova e 1 nota de trabalho. Escreva os seguintes métodos para esta classe:

media	calcula a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)
final	calcula quanto o aluno precisa para a prova final (retorna zero se ele não for para a final)

35. Escreva uma classe Data cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

construtor	define a data que determinado objeto (através de parâmetro), este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001
compara	recebe como parâmetro um outro objeto da Classe data, compare com a data corrente e retorne: • 0 se as datas forem iguais; • 1 se a data corrente for maior que a do parâmetro; • -1 se a data do parâmetro for maior que a corrente.
getDia	retorna o dia da data
getMes	retorna o mês da data
getMesExtenso	retorna o mês da data corrente por extenso
getAno	retorna o ano da data
isBissexto	retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário
clone	o objeto clona a si próprio, para isto, ele cria um novo objeto da classe Data com os mesmos valores de atributos e retorna sua referência pelo método

36. Escreva uma classe em que cada objeto representa um vôo que acontece em determinada data e em determinado horário. Cada vôo possui no máximo 100 passageiros, e a classe permite controlar a ocupaçãodas vagas. A classe deve ter os seguintes métodos:

construtor	configura os dados do vôo (recebidos como parâmetro): número do vôo, data (para armazenar a data utilize um objeto da classe Data, criada na questão anterior);
proximoLivre	retorna o número da próxima cadeira livre
verifica	verifica se o número da cadeira recebido como parâmetro está ocupada
ocupa	ocupa determinada cadeira do vôo, cujo número é recebido como parâmetro, e retorna verdadeiro se a cadeira ainda não estiver ocupada (operação foi bem sucedida) e falso caso contrário
vagas	retorna o número de cadeiras vagas disponíveis (não ocupadas) no vôo
getVoo	retorna o número do vôo
getData	retorna a data do vôo (na forma de objeto)
clone	o objeto clona a si próprio, para isto, ele cria um novo objeto da mesma classe e faz uma cópia dos valores de seus atributos

37. Considere a seguinte classe, cujo método respostaQuestao recebe como parâmetro o número de uma questão e retorna a sua resposta correta, proveniente de um gabarito.

```
public class Gabarito
{
  public char respostaQuestao(int numeroQuestao)
  {
    :
  }
}
```

Escreva uma classe classe Prova em que cada objeto representa uma prova feita por um aluno. Esta prova possui 15 questões de múltipla escolha (letras A a E). As 10 primeiras questões valem 0,5 ponto e as 5 últimas questões valem 1 ponto. Esta classe deverá controlar as questões respondidas pelo aluno. Para isto, a classe deve implementar os métodos:

construtor	recebe como parâmetro um objeto da classe Gabarito contendo o gabarito da prova
respostaAluno	recebe como parâmetro a resposta dada pelo aluno a uma questão; este método não recebe entre os parâmetros o número da questão, ele mesmo deve estabelecer um controle interno de modo que as questões sejam inseridas sequencialmente, ou seja, a primeira vez que o método é chamado, insere a primeira questão, a segunda, insere a segunda questão, e assim por diante.
acertos	retorna a quantidade de questões que o aluno acertou
nota	retorna a nota que o aluno tirou na prova
maior	recebe como parâmetro um outro objeto da classe Prova e retorna a nota do aluno que acertou mais questões; se houver empate, retorna a maior nota; se houver empate novamente, retorna a nota do aluno representado no objeto corrente

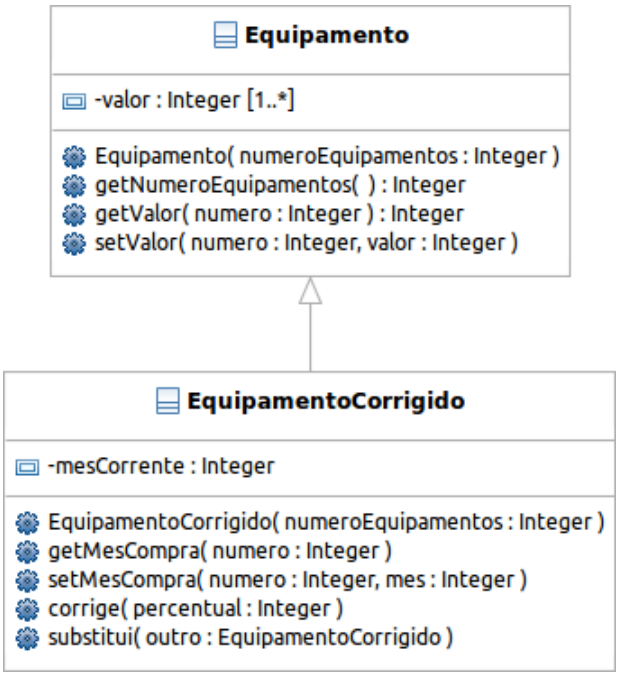
38. Escreva uma classe herdeira à voo criada na lista de exercícios anterior, que permita definir quantas cadeiras existem no máximo no voo e se dividir o avião em ala de fumantes e não fumantes. Para isto estaclasse deve acrescentar os atributos necessários e adicionar os seguintes métodos:

construtor	além dos parâmetros recebidos pelo construtor da superclasse, receberá também como parâmetros o número de vagas do voo e quantas cadeiras serão destinadas para fumantes
maxVagas	determina o número máximo de cadeiras no voo

cadeirasFumantes	determina quantas cadeiras estão destinadas aos fumantes (as demais serão automaticamente destinadas aos não fumantes); as cadeiras dos fumantes serão sempre as últimas do avião
tipo	recebe como parâmetro o número da cadeira e retorna 'F' se for uma cadeira para fumantes e 'N' se for para não fumantes

Os métodos proximoLivre, verifica e ocupa da superclasse devem ser adaptados para tratar o número máximo de vagas informado, ao invés do número fixo de 100.

39. Dada uma classe Equipamento na qual cada objeto representa um conjunto de N equipamentos de uma empresa com seus respectivos valores, cujo diagrama UML está representado a seguir:



construtor	recebe como parâmetro o número de equipamentos e cria um vetor de valores do respectivo tamanho
getNumeroEquipamentos	retorna o número de equipamentos
getValor	recebe como parâmetro o número do equipamento (começando de zero) e retorna seu valor
setValor	recebe como parâmetro o número do equipamento e seu valor e o registra

Cada equipamento possui um código numérico sequencial, começando de zero, que corresponde a sua posição no vetor.

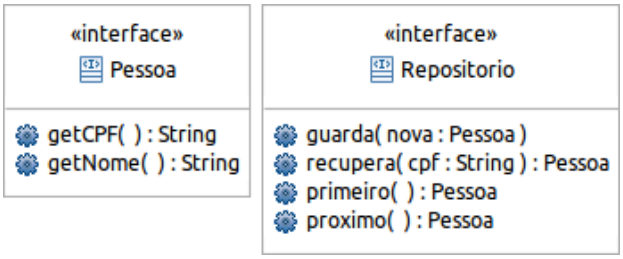
Escreva uma classe, herdeira da classe Equipamento, denominada EquipamentoCorrigido em que cada objeto representa os mesmos equipamentos com valor corrigido, conforme diagrama UML parcial representado anteriormente.

Todo equipamento só é corrigido anualmente no mês em que foi comprado, por este motivo a classe deve acrescentar para cada equipamento um registro do seu mês de compra. Além disto, deve possuir os métodos:

construtor	recebe como parâmetros o número de equipamentos e o mês corrente
getMesCompra	recebe como parâmetro o número do equipamento (começando de zero) e retorna seu mês de compra
setMesCompra	recebe como parâmetro o número do equipamento e seu mês de compra e o registra
corrige	este método recebe como parâmetro apenas o percentual de correção e corrige todos os equipamentos cujo mês de compra seja igual ao mês corrente; O objeto deve manter registrado em um atributo o mês corrente, que deve começar sempre em janeiro (quando o objeto é construído). Cada vez que este método é chamado, após a correção, o mês é incrementado de um e, se estiver em dezembro, retorna para janeiro
substitui	recebe como parâmetro um outro objeto da classe EquipamentoCorrigido e substitui o valor e o mês de compra de todos os equipamentos do objeto corrente pelos do objeto recebido como parâmetro; a operação só será realizada se ambos os objetos possuírem o mesmo número de equipamentos

Note que o atributo "valor" da classe Equipamento é privado, portanto, só poderá ser acessado indiretamente, até mesmo pela classe herdeira.

40. Dada as seguintes interfaces:



Pessoa – representa genericamente uma pessoa	
getCPF	retorna o CPF da pessoa
getNome	retorna o nome da pessoa
tipo	recebe como parâmetro o número da cadeira e retorna 'F' se for uma cadeira para fumantes e 'N' se for para não fumantes

Repositorio – representa genericamente um repositório	
guarda	guarda uma pessoa
recupera	recupera pessoa com o CPF informado

primeiro	se desloca para a primeira pessoa e a retorna
proximo	se desloca para a próxima pessoa e a retorna

Escreva uma classe denominada utilitários que possua os seguintes métodos:

duplica	Recebe como parâmetro dois objetos que implementam a interface Repositorio A e Be copia todas as pessoas do repositório A para o repositório B.
diferenca	Recebe como parâmetro três objetos que implementam a interface Repositorio A, B eC e coloque no repositório C todas as pessoas de A que não estiverem em B.

41. Um órgão de levantamento meteorológico possui equipamentos para medir a pluviosidade (pluviômetros), onde cada unidade é representada em um programa de computador por um objeto da classe:

```
public class Pluviometro
{
protected String tipo;
public Pluviometro(String tipo)
{
  :
}
public String getTipo()
{
  :
}
public int getPeso()
{
  :
}
public int getCapacidade()
{
  :
}
}
```

Construtor	Recebe como parâmetro o tipo de equipamento.
getTipo	Retorna o tipo do pluviômetro.
getPeso	Retorna o peso do pluviômetro em quilos. Este peso é calculado automaticamente pela classe a partir do tipo.
getCapacidade	Retorna a capacidade do pluviômetro em mililitros. Esta capacidade é calculada automaticamente pela classe a partir do tipo.

Os pluviômetros são carregados por caminhões que, no programa de computador, são representados genericamente por objetos da classe `Caminhao` (esta classe não deve ser implementada nesta questão). A classe define os seguintes métodos:

Construtor	Recebe como parâmetro a quantidade de equipamentos que o caminhão irá carregar.
inserePluviometro	Recebe como parâmetro um objeto da classe `Pluviometro` e o coloca dentro do caminhão se a capacidade do mesmo permitir.

Cada objeto da classe representa um caminhão. Esta classe não possui nenhum atributo e seus métodos

não possuem implementação, pois serão implementados nas subclasses.

Escreva duas classes herdeiras da classe `Caminhao` que representam dois tipos deste veículo com capacidades diferentes:

Caminhão Alfa

Consegue carregar no máximo 5 toneladas de pluviômetros, independente da quantidade e tipo.

Caminhão Beta

Consegue carregar qualquer quantidade e peso de pluviômetros, no entanto, não é capaz de carregar mais do que um pluviômetro de cada tipo.

Ambas as classes devem sobrescrever o método `inserePluviometro`.

42. Dada uma classe denominada `Controle`, definida em um sistema de controle de transporte de pluviômetros, que possui dois métodos estáticos:

`leString()`	Solicita ao usuário uma String pelo teclado e retorna através do método (função tipo String).
`leInteito()`	Solicita ao usuário um valor inteiro pelo teclado e retorna através do método (função tipo int).

Escreva uma classe herdeira de `Controle` que acrescente um método estático responsável pela seleção do caminhão mais apto a distribuição de pluviômetros. O caminhão será aquele capaz de conduzir pluviômetros cuja soma de capacidade seja a maior, independente do tipo de cada um deles.

O programa irá solicitar uma lista de caminhões. Para cada caminhão ele pergunta:

- Tipo do caminhão (‘Alfa’ ou ‘Beta’);
- número de pluviômetros a ser transportados;
- lista dos pluviômetros a ser transportados no caminhão (para cada pluviômetro é digitadoapenas seu tipo).

A lista encerra quando é digitado ‘Fim’ para o Tipo do caminhão.

Depois de digitada toda a lista o programa imprime na tela (`System.out`) os seguintes dados do caminhão mais apto:

- tipo do caminhão (Alfa ou Beta);
- lista dos pluviômetros transportados no caminhão.

Em ambas as questões, devem ser criados os atributos que se fizerem necessários.

43. Dada a seguinte classe cujos objetos representam uma saldo bancário (armazenado no atributo valor)e sua operação de movimentação da conta:

```
public class Saldo implements Transacao {

    private float valor;

    public float getValor() {

        return valor;

    }
    public void movimento(float valorMovimento) {
```

A classe implementa a seguinte interface:

```
public interface Transacao {

    public void movimento(float valorMovimento);
```

Escreva uma classe herdeira de `Saldo`, denominada `SaldoRepl`, que possibilite replicar suas operações de movimento em outros objetos.

Qualquer objeto `X` da classe `SaldoRepl` deve oferecer um método em que outros *objetos assinantes* se cadastrem. Este método será acionado pelos *objetos assinantes* que desejam ser avisados quando ocorrer uma operação de movimento em `X`. A única exigência é que os *objetos assinantes* devemimplementar a interface `Transacao`. A operação de `movimento` da classe `SaldoRepl` deverá ser estendida de tal modo que, cada vez que ela é acionada em `X`, este mesmo objeto `X` irá acionar a mesma operação (com o mesmo parâmetro) de todos os *objetos assinantes*.

44. Um laboratório de pesquisa está automatizando seu processo de testes biológicos. Cada experimento envolve uma sequência de ações automáticas a ser aplicadas em uma cultura de microrganismos, a fim de verificar a reação dos mesmos. Exemplos de ações são: elevação da temperatura ambiente, aplicação de um produto químico etc.

Cada ação é aplicada automaticamente por um robô controlado por um objeto de software que implementa a interface:

```
public interface RoboAction {
    public void execute();
}
```

Cada robô executa uma única ação e não é necessário informar parâmetros da ação.

Escreva uma classe denominada Experimento cujos objetos são capazes de aplicar um teste envolvendo uma sequência de ações, que podem ser realizadas por diferentes robôs. Cada objeto Experimento deve guardar uma lista com a sequência de ações a ser executada (cada ação é executada porum único objeto robô o qual ele guarda a referência) e deve implementar operações que permitam adicionar ações (sempre no final da sequência) e executar o teste (na sequência em que as ações foram cadastradas).

Depois de ter resolvido o exercício, leia sobre o design pattern Command (http://en.wikipedia.org/wiki/Command_pattern) e analise qual a relação com o código que vocêimplementou.

45. O laboratório resolveu criar um processo automático para descobrir experimentos, que são mais eficientes na destruição de um dado tipo de microrganismo. Como as possibilidades de combinação são infinitas, decidiu-se usar um método baseado na programação genética.

Considere que existe uma classe ExperimentoMonitorado representando um experimento com funcionalidades extra de monitoramento. Esta classe é herdeira de Experimento e acrescenta o método contaMicro(), que não recebe parâmetros e retorna a contagem de microrganismos. Este método pode ser aplicado antes e depois da execução do experimento, de modo a verificar sua efetividade.

Implemente o processo em duas etapas:

Crie uma variante do método clone chamado cloneMutant que produz mutantes. Este método ao realizar a clonagem aplica uma única mutação na cópia que pode ser:

- um par de operações trocadas
- uma operação ausente
- uma operação duplicada

A posição da operação que sofre mutação também deve ser aleatória.

Crie um método estático (apenas o método sem uma classe) que receba dois parâmetros: um objeto da classe Experimento (base) e o número de ciclos que será usado para encontrar o melhor experimento. Este método deve atuar da seguinte maneira:

(1) cria uma lista de melhores experimentos com 5 posições (inicialmente vazia)

(2) adiciona o experimento base (recebido como parâmetro) na lista dos melhores

(3) percorre a lista dos melhores clonando (com mutação) cada um de seus componentes

(4) aplica todos os experimentos disponíveis (os da lista dos melhores e os clones)

(5) seleciona os 5 melhores resultados (se houver) e os coloca na lista dos melhores

(6) conta um ciclo e retorna ao passo (3) se não tiver completado o número de ciclos

46. Dada uma classe abstrata (já implementada) denominada `Noticiario` cuja função é enviar notícias para assinantes. Cada objeto desta classe fica monitorando uma fonte de notícias e quando surge umanova notícia chama um método abstrato da própria classe com a seguinte assinatura:

```
public abstract void notificaNoticia(String textoNoticia, int dia, int mes, String topico);
```

Os parâmetros indicam o texto da notícia, seu dia e mês e o tópico a que se refere respectivamente. Tópico é uma string usada para classificar as notícias por tema. Exemplos de tópico: lançamento de livro, greve, inovação tecnológica etc.

Considerando que qualquer objeto consumidor de notícias implemente a interface:

```
public interface ConsomeNoticia {
    public void notificaNoticia(String textoNoticia, int dia, int mes, String topico);
}
```

Escreva uma classe herdeira de `Noticiario` chamada `NoticiarioAssina` para divulgar suas notícias.

47. Implemente as classes de dois possíveis tipos de objeto que consomem esta notícia (implementam

`ConsomeNoticia`): agregadores e publicadores.

Um objeto publicador publica cada notícia recebida (neste exercício, imprime no console).

Um objeto agregador concatena várias notícias recebidas e as envia para assinantes, implementando omesmo *pattern* do `NoticiarioAssina`. Ele pode ser de dois tipos: agregador por tópico ou agregador pormês.

O agregador por tópico escolhe um tópico para agregar (informado no construtor) e concatena sempredez notícias do mesmo tópico. Cada vez que consegue agregar dez notícias, ele envia a string concatenadapara seus assinantes com o dia e o mês da última notícia.

O agregador por mês concatena todas as notícias do mesmo mês e, apenas quando chega uma notícia de um mês diferente, envia a string concatenada para seus assinantes com o dia zero e o tópico "mensal".

48. Faça uma aplicação que concatene agregadores em sequência, ligados a um objeto `NoticiaAssina`, com um publicador no final. Uma concatenação interessante envolve o filtrar um tópico e o agregá-lo mensalmente.

49. Considere que uma biblioteca gráfica disponibiliza uma classe `Visual` os seguintes métodos estáticos:

`drawLine(x1, y1, x2, y2)`	Desenha uma linha. x1 e y1 – coordenadas do ponto de origem; x2e y2 – coordenadas do ponto de destino.
`drawRectangle(x, y, altura, largura)`	Desenha um retângulo. x e y – coordenadas do canto esquerdosuperior; altura e largura do retângulo.
`drawCircle(x, y, raio)`	Desenha um círculo. x e y – coordenadas do centro; raio – raio docírculo.

A partir desta biblioteca:

a) Escreve uma classe que seja capaz de armazenar o estado e plotar um objeto geométrico.

b) Considere a seguinte estratégia padronizada para definir as medidas de uma reta, um retângulo e um círculo: para os três casos é definido um *bounding box*, ou seja, um retângulo delimitador com as coordenadas x1, y1, x2, y2.

- a linha é desenhada como uma diagonal partindo de x1, y1 até x2, y2;
- o retângulo tem seu canto esquerdo superior em x1, y1 e o esquerdo inferior em x2, y2;
- o círculo corresponde ao maior círculo que pode ser desenhado dentro do retângulo.

50. Crie uma classe que modele uma bola e permita trocar e consultar a cor da bola.

51. Crie uma classe que modele um quadrado e permita definir , alterar e consultar o tamanho dos lados e obter a área.