

## **Fundamentos do Java**

### **Lógica de Programação**

1. Escrever um programa que apresente no vídeo seu nome completo em uma linha e o RG na outra.
2. Um programa contém duas variáveis inteiras e cada uma destas variáveis deve ter um valor fornecido pelo usuário. Escrever um programa que mostre os seguintes resultados:
  - a. A soma das duas variáveis
  - b. A diferença entre as duas variáveis
  - c. O dobro da primeira mais o triplo da segunda variável
  - d. A multiplicação das duas variáveis
3. Escrever um programa que contenha duas variáveis reais, onde a primeira represente um valor e a segunda, uma porcentagem, ambos fornecidos. O programa deve mostrar:
  - a. 3.1. O valor da primeira variável
  - b. 3.2. O valor da porcentagem com seu símbolo
  - c. 3.3. O resultado da aplicação da porcentagem sobre o valor da primeira variável
4. Escrever um programa que declare duas variáveis inteiras e mostre no vídeo o valor das duas e em seguida, a maior delas.
5. Uma folha de pagamento simplificada contém os itens: Nome do Funcionário, Salário e Taxa de Desconto de INSS. O nome e o salário devem ser fornecidos pelo usuário. Criar um programa que considere uma taxa de desconto de 8% e mostre no vídeo:
  - a. nome do Funcionário
  - b. salário bruto
  - c. valor do INSS
  - d. salário líquido
6. Escrever um programa que apresente seu nome no vídeo 50 vezes.
7. Escrever um programa que incremente uma variável com valor mínimo 0 e valor máximo 100 e as mostre no vídeo.
8. Considerando o problema anterior, criar um novo programa que incremente uma variável com a mesma faixa de valores, e mostre no vídeo somente os números pares ou múltiplos de 6.
9. Criar um programa que leia uma base e um expoente e, em seguida, mostre o valor da potência.

10. Escrever um programa que mostre no vídeo a tabuada do 2, do 3 e do 5
11. Escrever um programa que calcule o fatorial de um número inteiro fornecido pelo usuário.
12. Elaborar um programa que mostre os N primeiros termos da série de Fibonacci (N deve ser lido do teclado). Esta série é definida pela seguinte regra: A partir do terceiro elemento, cada termo é a soma dos dois anteriores

1        1        2        3        5        8        13 ...

13. Criar um programa que incremente uma variável inteira iniciando em 0 e terminando em 50, mostrando os valores múltiplos de 8, o número da ocorrência em cada exibição e quantas vezes o múltiplo ocorreu.
14. Escrever um programa que incremente uma variável iniciando em 0 e terminando em 200. A cada passagem pela variável, se o valor for par, este deverá ser acumulado em uma variável e, se for ímpar, acumulado em outra.
15. Escrever um programa que considere uma variável inteira, cujo valor é fornecido pelo usuário, e mostre sua representação binária (32 bits).
16. Sabe-se que a base do logaritmo neperiano é representada por e, e vale 2,718281 aproximadamente:

$$e = 2.718281...$$

Como é um número irracional, pode ser calculado pela expressão:

$$e = \lim_{n \rightarrow \infty} \left( 1 + \frac{1}{n} \right)^n$$

Considerando n = 100, 500, 1000 e 2000, calcular o valor de e, verificando e comparando os resultados.

17. Se a taxa de juros compostos de uma determinada aplicação vale 1,25% ao mês, calcular a taxa equivalente em um intervalo de meses fornecido pelo usuário (número de meses). Se um valor de R\$1.000,00 for aplicado com esta taxa, qual será o valor final após um ano?
18. Escreva uma classe chamada Funcionario. Nela, definir como dados membro (atributos): nome, idade, salário e cargo. Em seguida, criar:
  - a. Um método capaz de mostrar os dados dos objetos desta classe, incluindo a informação sobre a maioridade ou não
  - b. Um método que receba como parâmetros os atributos da classe
19. Uma classe chamada Conta possui os atributos: cliente, número da conta, tipo de conta (1-simples, 2-especial), limite de crédito e saldo. Escrever esta classe, juntamente com os métodos:
  - a. saque() : recebe como parâmetro um número real e o desconta do saldo
  - b. deposito() : recebe como parâmetro um número real e o acrescenta ao saldo

- c. `extrato()`: apresenta no vídeo os dados do cliente, incluindo seu saldo disponível, seu limite e o saldo total (se a conta for especial)

Notas:

- O saldo total não deve ser negativo (incluindo o limite de crédito)
  - Os valores do saque e do depósito não devem ser negativos. Mesmo que o usuário forneça um valor negativo, convertê-lo para positivo
  - O limite de crédito só deve ser atribuído ao cliente se a conta for especial
20. (Geometria) Criar uma classe chamada `Ponto`, que represente informações de um plano cartesiano. A classe deve conter como atributos as coordenadas  $x$  e  $y$ . A partir destas informações:
- a. Criar um método capaz de receber como parâmetro valores para as coordenadas
  - b. Um método para exibir os valores das coordenadas de um ponto
  - c. Um método que receba dois parâmetros, representando coordenadas de um ponto, os compare com as coordenadas do ponto que executa o método, e apresente uma mensagem no vídeo, de acordo com as informações a seguir:
    - se os dois pares de coordenadas formam uma reta, ou se coincidem
    - se formarem uma reta, se esta reta forma ângulo de 45 graus com o eixo  $x$
    - a projeção desta reta (se for o caso) sobre o eixo  $x$  e sobre o eixo  $y$
21. Em um parque de diversões, cada pessoa paga ingressos de acordo com a tabela abaixo:

Idade	Valor do Bilhete
Inferior a 6 anos	isento
Entre 6 e 12 anos	15,00
de 13 a 65 anos	30,00
acima de 65 anos	25,00

Cada pessoa deve representar um objeto de uma classe chamada `Pessoas`. Escrever esta classe com os atributos: nome, idade e valor. Nesta classe, escrever um método para receber adequadamente estas informações, e outro para exibi-las na tela.

22. O índice de massa corpórea (IMC) de uma pessoa é calculado pela fórmula:

$$IMC = \frac{peso}{altura^2}$$

De acordo com o valor obtido, deve-se determinar a classificação de uma pessoa, a saber:

- IMC abaixo de 17: pessoa magra em excesso
- de 17 a 25 : boa forma física
- acima de 25, até 30 : levemente obeso
- acima de 30: obeso.

23. Escrever uma classe chamada IMC, contendo como dados membro o nome da pessoa, a altura e o peso. Os dados devem ser lidos através do teclado. Deve haver também um método para apresentar a classificação de uma pessoa, de acordo com estes dois valores.
24. Escreva um programa em Java que leia um valor inteiro (o número deve ter quatro dígitos: 1000 - 9999). O seu programa deverá imprimir a quantidade de dígitos que são pares. Por exemplo: suponha que o valor de entrada seja 9679. A quantidade de dígitos pares é igual a 1.
25. Escreva um programa em Java que leia vários números inteiros e positivos. O seu programa deverá imprimir a quantidade de números que são maiores do que 5. A condição de parada é quando o usuário digitar o número 0 (zero).
26. Escreva um programa em Java que gere e imprima alguns termos da série de Fibonacci. A quantidade de termos deverá ser informada pelo usuário. Série: 0, 1, 1, 2, 3, 5, 8, ...
27. (Algoritmo numérico) Uma das formas de se calcular a raiz quadrada de um número positivo  $y$  é usando o seguinte algoritmo:

i) a primeira aproximação para a raiz de  $y$  pode ser calculada como:  $x_1 = \frac{y}{2}$

ii) as sucessivas aproximações serão:  $x_{n+1} = \frac{x_n^2 + y}{2x_n}$

Escreva um programa em Java que calcule a raiz quadrada de um número inteiro e positivo usando o roteiro especificado acima. O seu programa deverá executar 25 aproximações.

28. O número 3025 apresenta a seguinte característica:  $30+25=55$  e  $55^2=3025$ . Escreva um programa em Java que encontre todos os números em Java ( $>999$  e  $< 10000$ ) que apresentam tal característica.

## **Orientação a Objetos**

1. Para a situação abaixo identifique as possíveis classes, atributos e métodos. O cenário se refere às informações contidas em uma conta de energia elétrica.

Para cada conta de luz cadastra-se: data em que a leitura do relógio de luz foi realizada, número da leitura, quantidade de Kw gasto do mês, valor a pagar pela conta, data do pagamento e média de consumo.

Mensalmente, são realizadas as seguintes pesquisas: mês de menor consumo e maior consumo.

2. Sabe-se que os números complexos têm diversas aplicações na engenharia. Um número complexo é formado pela combinação de dois números reais chamados de parte real e parte imaginária. Dentre as principais operações com números complexos podemos citar: soma, subtração, multiplicação e divisão.
3. Um determinado sistema faz o controle dos funcionários e dos clientes de uma loja. Tanto o cliente quanto o funcionário apresentam algumas características em comum tais como: um código numérico único, nome completo, endereço, cpf e rg. Além dessas informações, para o cliente também é necessário o limite de crédito e para o funcionário o valor de seu salário.

Para esse cenário apresente as possíveis classes envolvidas. Destacar também os atributos e métodos envolvidos.

4. Considere uma instituição de ensino que ministra aulas para o ensino fundamental, médio e universitário. Defina um cenário para essa situação. Defina também atributos e métodos para cada classe presente no cenário definido.
5. Considere as seguintes classes:

Funcionário
matrícula: inteiro nome: string dataNascimento: date salário: real dataAdmissao: date
cadastrar() obterIdade() reajustarSalario(percentual: real)

Cliente
código: string nome: string dataNascimento: date
cadastrar() obterIdade()

Remodele as classes acima de forma a usar a herança

## Programação de Classes e Métodos

1. Escreva uma classe em Java chamada Fone. Esta classe deve apresentar três atributos: fone residencial, fone comercial e fone celular. Codificar o método construtor para inicializar os atributos e também um método para retornar todos os números de telefone em uma mesma variável. Escrever um programa em Java para testar a classe Fone.
2. Crie uma classe chamada Retângulo. A classe deve ter apenas duas variáveis de instância: largura e altura. A classe deve ter o método construtor, um método para calcular o perímetro (soma dos lados), e outro para calcular a área (largura x altura) e também métodos para alterar os valores da altura e da largura e também para retornar esses valores. Escreva um programa para testar a classe Retângulo.
3. Codificar a classe Hora. A classe tem 3 variáveis de instância: hora (0-23), minuto (0-59) e segundo (0-59). A classe deverá apresentar um construtor para inicializar os atributos. Caso os valores informados estejam fora

da faixa especificada, as variáveis deverão ser inicializadas com 0. A classe também deverá apresentar um método para retornar a hora no formato xx:xx:xx. Escreva um programa para testar a classe Hora.

4. Codificar uma classe chamada ContaPoupança. A classe deverá ter dois atributos: atributo static chamado taxaDeJurosAnual e outro chamado saldoPoupança. A classe deverá apresentar um método chamado calcularJurosMensais() que atualiza o saldo de acordo com a taxa de juros anual (taxa dividida por 12), um método static chamado atualizarTaxaDeJuros() para alterar a taxa de juros anual com um novo valor.
5. Escreva um programa para testar a classe ContaPoupança. Instanciar dois objetos da classe com saldos R\$ 2000,00 e R\$ 3000,00, respectivamente. Configurar a taxa anual de juros para 4%, calcular os juros mensais e imprimir os saldos de cada conta. Atualizar a taxa de juros para 5%, calcular os novos juros mensais e imprimir os saldos.

## Manipulação de Arrays

1. Qual a saída do programa abaixo (não implementar, apenas analisar o resultado)?

```
public class Exercicio1 {
    public static void main(String args[]) {
        int x[] = {1, 10, 5, 6, 7, 2, 3, 9, 4, 8};
        metodo1(x);
        metodo2(x);
        metodo3(x);
    }

    public static void metodo1(int y[]) {
        int i, j, aux;
        for(i = 0; i < y.length; i++)
            for(j = 1; j < y.length; j++)
                if(y[j] < y[j-1])
                {
                    aux = y[j];
                    y[j] = y[j-1];
                    y[j-1] = aux;
                }
    }

    public static void metodo2(int z[]) {
        int aux, i;
        for(i = 0; i < z.length/2; i++)
        {
            aux = z[i];
            z[i] = z[z.length-1-i];
            z[z.length-1-i] = aux;
        }
    }

    public static void metodo3(int x[]) {
        for(int i : x)
            System.out.print(i+"\t");
    }
}
```

2. Qual a saída do programa abaixo (não implementar, apenas analisar o resultado)?

```
public class Exercicio2 {
    public static void main(String[] args) {
        int x = metodo(new int[]{1, 2, 3, 4, 5, 6, 7, 8, 9, 10});
        System.out.println(x);
    }

    public static int metodo(int t[])
    {
        int cont = 0;
        for(int i = 0; i < t.length; i++)
            if(t[i]%2 == 0 && t[i]%4 == 0)
                cont++;
        return(cont);
    }
}
```

3. Qual a saída do programa abaixo (não implementar, apenas analisar o resultado)?

```
public class Exercicio3 {
    public static void main(String[] args) {
        int x = metodo(new int[][]{{1, 2, 3}, {4, 5, 6}, {7, 8, 9, 10}});
        System.out.println(x);
    }

    public static int metodo(int y[][]) {
        int i, j, m1, m2;
        m1 = m2 = y[0][0];
        for(i = 0; i < y.length; i++) //determina o número de linhas da matriz
            for(j = 0; j < y[i].length; j++) //determina o núm de colunas de cada linha
            {
                if(i == j)
                    if(y[i][j] > m1)
                        m1 = y[i][j];

                if(i+j == y.length-1)
                    if(y[i][j] > m2)
                        m2 = y[i][j];
            }
        return(m1-m2);
    }
}
```

4. Escreva um programa em Java que declare uma matriz quadrada de ordem 10 (no método main()). Um método deverá preencher a matriz com números aleatórios e um outro método deverá calcular e retornar o número minimax. O número minimax é o menor número da linha em que se encontra o maior número da matriz.
5. Escreva um programa em Java que alocue uma matriz quadrada de acordo com a dimensão informada pelo usuário. A matriz deverá ter no mínimo ordem 2 e no máximo ordem 12. Um método deverá também preencher a matriz com valores aleatórios (o limite também deverá ser informado pelo usuário). Um segundo método deverá retornar a diagonal principal e a diagonal secundária, que deverá ser impressa no main().
6. Escreva uma classe chamada Aluno tendo como atributos o nome do aluno, e as notas 1 e 2. A classe deverá ter o método construtor, método para calcular a média aritmética das notas e também um método para

retornar o nome, as notas e a média (tudo junto!!). Escreva um programa que instancie 5 objetos do tipo aluno e os armazene em um vetor. Imprima todas as informações dos alunos em ordem crescente de média.

7. Escreva uma classe chamada Círculo contendo os seguintes atributos: coordenadas x e y do centro do círculo e o valor do raio. Codificar o método construtor e também um método para calcular a área do círculo ( $2\pi Raio^2$ ). Escreva um programa em Java que instancie 5 objetos e os armazene em um vetor. Ordenar (ordem crescente) os objetos em relação à área. Imprima todas as informações dos os objetos no vídeo. Observação: para o valor de “pi” usar: Math.PI.
8. Tendo como base a classe Complexo criada nas aulas, escreva um programa em Java que crie e armazene em um vetor uma quantidade de objetos definidas pelo usuário. Um método deverá retornar o objeto que apresenta o maior valor na parte real do número. Caso dois ou mais objetos tenham o mesmo valor na parte real, todos os objetos deverão ser retornados e impressos no vídeo.
9. **(Exercício teórico)** O método abaixo está codificado corretamente? Caso não esteja aponte o erro e corrija.

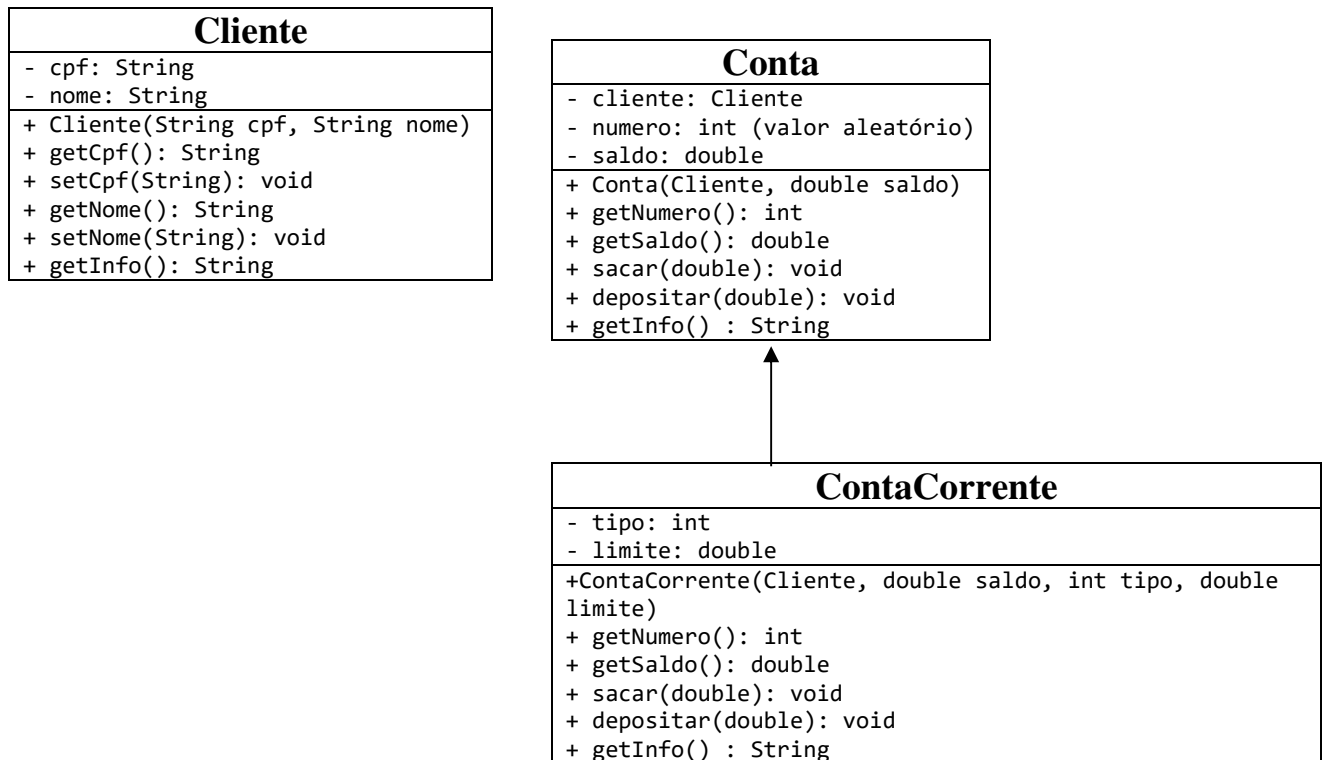
```
public boolean metodo1(int...x, int y)
{
    boolean aux = false;
    for(int i = 0; i < x.length; i++)
        if(x[i] == y)
        {
            aux = true;
            break;
        }
    return(aux);
}
```

10. Escreva apenas um método que receba como parâmetro um valor inteiro e também uma lista de parâmetros variável do tipo inteiro. Seu método deverá retornar o número de vezes que o número está contido na lista.
11. **(Exercício desafio)** Escreva apenas um método que receba como parâmetro uma lista de parâmetros variável do tipo inteiro. O seu método deverá percorrer a lista e eliminar as duplicações dos números, ou seja, cada número só pode aparecer uma única vez. O método deverá retornar a nova lista.

## Encapsulamento e Herança

O diagrama de classes abaixo representa uma aplicação bancária. Codifique cada uma das classes abaixo.





#### Observações:

- O atributo tipo na classe ContaCorrente especifica o tipo da conta. 1 (conta comum) ou 2 (conta especial). Se a conta for comum, o limite é zero.

## Herança e Polimorfismo

1. Escreva uma hierarquia de herança para as classes Quadrilátero, Trapézio, Paralelogramo, Retângulo e Quadrado. Utilize Quadrilátero como a superclasse (abstrata) da hierarquia. Especifique as variáveis de instância (todas *private*) para cada classe. Codifique também uma interface chamada Calcular contendo os métodos: calcularArea() e calcularPerimetro(). Todas as classes da hierarquia deverão implementar a interface Calcular.
2. Codificar a classe principal. Gerar objetos de todas as classes (exceto da superclasse) e armazenar em um vetor do tipo Quadrilátero.
3. Escrever um método estático (imprimirArea()) que receba como parâmetro o vetor do tipo Quadrilátero e imprima o tipo do quadrilátero (Retângulo, Quadrado, etc..) e também a sua área.
4. Escrever um método estático (imprimirPerimetro()) que receba como parâmetro o vetor do tipo Quadrilátero e imprima o tipo do quadrilátero e também o seu perímetro.

## Exercícios Complementares

1. Qual o resultado da execução das classes abaixo?

```
public class Clidders
{
    public final void flipper()
    {
        System.out.println("Clidder");
    }
}

public class Clidlets extends Clidders
{
    public void flipper()
    {
        System.out.println("Flip a Clidlet");
        super.flipper();
    }

    public static void main(String args[])
    {
        new Clidlets().flipper();
    }
}
```

2. Dado:

```
public interface Frobnicate {
    public void twiddle(String s);
}
```

Quais são corretas? Marque todas.

- a)
- ```
public abstract class Frob implements Frobnicate {
    public abstract void twiddle(String s) { }
}
```
- b)
- ```
public abstract class Frob implements Frobnicate { }
```
- c)
- ```
public class Frob extends Frobnicate {
    public void twiddle(Integer i) { }
}
```
- d)
- ```
public class Frob implements Frobnicate {
    public void twiddle(Integer i) { }
}
```
- e)
- ```
public class Frob implements Frobnicate {
    public void twiddle(String i) { }
    public void twiddle(Integer s) { }
}
```

3. Analise as classes abaixo e indique a saída após a execução.

Neste exercício no lugar de *public* é *private* senão não funciona

```
public class Clidder
{
    private final void flipper(){
        System.out.println("Clidder");
    }
}

public class Clidlet extends Clidder
{
    public final void flipper()
    {
        System.out.println("Clidlet");
    }

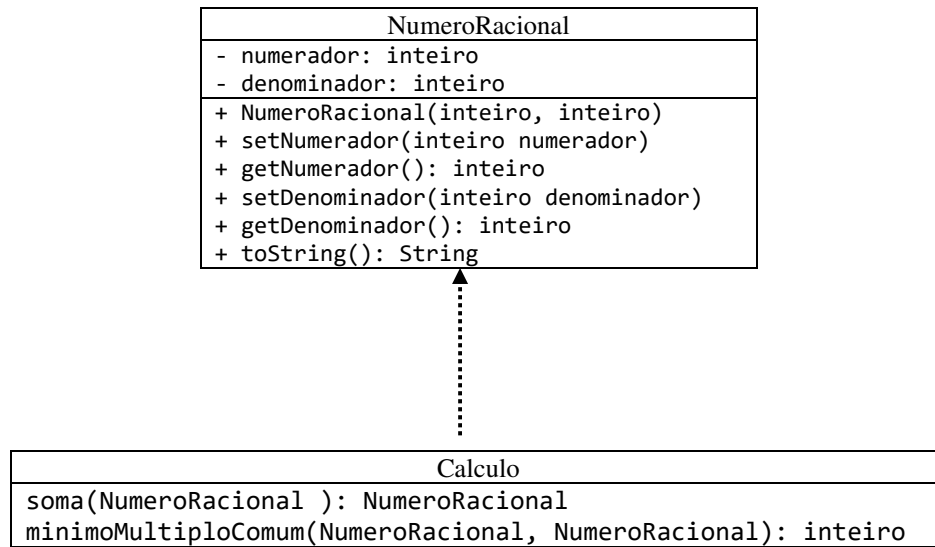
    public static void main(String args[])
    {
        new Clidlet().flipper();
    }
}
```

4. Analise as classes abaixo e indique a saída a após a execução.

```
class Uber
{
    static int y = 2;
    Uber(int x)
    {
        this();
        y = y * 2;
    }
    Uber()
    {
        y++;
    }
}
```

```
class Mirror extends Uber
{
    Mirror()
    {
        super(y);
        y = y + 3;
    }
    public static void main(String args[])
    {
        new Mirror();
        System.out.println(y);
    }
}
```

5. Escreva apenas um método que receba como parâmetro um vetor de tipo genérico. O método deverá retornar o tamanho do vetor.
6. Escreva apenas um método que receba como parâmetro dois vetores de tipo genérico. O método deverá retornar um novo vetor que é composto pela intersecção dos dois vetores recebidos como parâmetro. A intersecção é formada pelos elementos comuns aos dois vetores (sem repetição).
7. Considere uma classe chamada Aluno contendo os seguintes atributos *private*: nome, p1, p2. A classe apresenta o método construtor, método para calcular e retornar a média aritmética e também os métodos *setters* e *getters* para cada atributo. Pede-se:
- Codificar a classe;
  - Escreva um programa (método main()) para testar a classe Aluno. Seu programa deverá gravar objetos do tipo Aluno em um ArrayList<Aluno>. Um método deverá ser chamado para imprimir os nomes dos alunos em ordem crescente de média aritmética.
8. Escreva um método que receba como parâmetro um vetor estático contendo objetos do tipo Aluno (exercício 7). O método deverá imprimir o nome de todos os alunos que apresentam a maior média da classe.
9. Considere a seguinte hierarquia de classes:



- Codificar as classes.
- Escreva um programa (método `main()`). Instancie dois objetos do tipo `NumeroRacional`. Imprima no vídeo a soma dos números e também o valor do MMC (mínimo múltiplo comum).
- Escreva apenas um método que receba como parâmetro um vetor estático do tipo `NumeroRacional`. O método deverá retornar o objeto `NumeroRacional` que apresenta o maior valor no atributo `denominador`.
- Reescreva o método anterior supondo que os objetos estejam armazenados em um `ArrayList<NumeroRacional>` ao invés de um vetor estático.