

Aluno(a): _____

Turma: _____

Data: _____

Faça distinção clara entre maiúsculas e minúsculas.
Códigos desnecessários e que reduzam o desempenho do sistema serão penalizados.
Utilize as boas práticas de programação.
LEIA AS QUESTÕES ATÉ O FINAL ANTES DE COMEÇAR.
Sempre que for possível, declare os atributos de instância privados.

Questão 1 (5.5) – Resolva:

1. Crie uma classe *abstrata* chamada Livro conforme o diagrama abaixo:

<i>Livro</i>
- nome:String - autor: String - ano: int
+Livro(ano: int) +getters e setters

- Crie também, na classe acima, o atributo ISBN do livro. Crie esse atributo como privado e do tipo String. Crie os getters e setters para ele. Sobrescreva o método de object que permite saber se dois objetos do tipo Livro são iguais. Considere dois livros iguais se o ISBN de ambos for o mesmo.
2. Crie uma classe concreta chamada LivroDeLivraria que seja filha de Livro.
- Crie um novo atributo do tipo booleano chamado vendido. Crie também os getters e setters para esse atributo.
 - Crie um construtor que receba dois argumentos: *ano* e *vend*, que servirão para inicializar os valores dos atributos *ano* e *vendido*. Preste atenção!
3. Crie uma classe concreta chamada LivroDeBiblioteca que seja filha de Livro.
- Crie um novo atributo do tipo booleano chamado *alugado*. Crie também os getters e setters deste atributo.
 - Crie um construtor que receba dois argumentos: *ano* e *alug*, que servirão para inicializar os valores dos atributos *ano* e *alugado*. Preste atenção!
4. Crie a classe abaixo:

<i>Instituição</i>
- nome:String - listaDeLivros: ArrayList
+setNome(n: String): void +getNome() : String ...

- a. Você precisará criar mais dois métodos. Um para inserir livros na lista e outro para retornar a lista de livros. Mais tarde, você irá precisar desses métodos na classe Principal.

5. Crie uma classe chamada Principal que contenha o método main. Crie 2 livros do tipo LivroDeBiblioteca e 2 livros do tipo LivroDeLivraria. Atribua valores a todos os atributos de instância.

- a. No método main, crie uma instância da classe Instituição. Adicione os 4 livros criados à instituição. Chame o método imprimeRelatorioGeral que será criado a seguir. Não instancie um ArrayList no método main, utilize sempre o que existe na classe Instituição.
- b. Crie um método chamado imprimeRelatorioGeral que receba como argumento uma lista de livros (Nessa lista estarão objetos do tipo LivroDeLivraria e LivroDeBiblioteca). Faça a iteração (com Iterator) pela lista para:
 - i. Caso seja um LivroDeBiblioteca, imprimir no console o nome, o ano e se este livro está alugado ou não.
 - ii. Caso seja um LivroDeLivraria, imprimir no console o nome, o ano e se este livro foi vendido ou não.
- c. Exiba no console a seguinte mensagem: "Qual o ISBN do livro a ser consultado?". Receba do console uma String que representa o ISBN. Verifique se na Instituição criada existe um livro com esse ISBN. Imprima o nome, ano e autor do livro com o ISBN consultado. Nesse item não é permitido iterar pela lista de livros da Instituição, ou seja, não use for, while, Iterator, etc.

Questão 2 (0.5) – Resolva:

Crie uma interface chamada corredor com um método chamado calcularVelocidade que não retorne nada e não contenha parâmetros. Crie uma classe chamada Carro que implemente essa interface. Faça o que for necessário para que a interface e a classe compilem.

Questão 3 (2) – O que sai no console?

```
public class Escreve {  
    public Escreve() {  
        System.out.println("F");  
    }  
}  
  
public class Caneta extends Escreve {  
    private String cor;  
    private static int quantidade;  
    public Caneta() {  
        System.out.println("X");  
    }  
    public String getCor() {  
        return cor;  
    }  
    public void setCor(String cor) {  
        this.cor = cor;  
    }  
    public int getQuantidade() {  
        return quantidade;  
    }  
    public void setQuantidade(int quantidade) {  
        Caneta.quantidade = quantidade;  
    }  
}
```

```

public class Principal {
    public static void main(String[] args) {
        Caneta c = new Caneta();
        c.setCor("azul");
        Caneta c2 = new Caneta();
        c2.setCor("preta");
        c2.setQuantidade(3);
        metodoCan1(c);
        metodoCan2(c2);
        int i=15;
        Integer j = i++;
        qtd(i);
        ++i;
        qtd2(j);
        System.out.println(i);
        System.out.println(j);
        System.out.println(c.getQuantidade());
        System.out.println(c2.getQuantidade());
        System.out.println(c.getCor());
        System.out.println(c2.getCor());
    }
    public static void metodoCan1(Caneta c) {
        c.setCor("vermelho");
        c.setQuantidade(9);
    }
    public static void metodoCan2(Caneta c) {
        c.setQuantidade(2);
        c = new Caneta();
        c.setCor("lilas");
    }
    public static void qtd(int i) {
        i = 19;
    }
    public static Integer qtd2(Integer i) {
        return i+15;
    }
}

```

Questão 4 (2) – Resolva:

Crie uma classe chamada MeuNumero com um método estático chamado media. Esse método deve receber um argumento do tipo String. Essa String sempre será no formato número#número#... (ex: 15#14#3...). Você não sabe quantos caracteres '#' virão na String, mas sabe que um número está sempre entre eles. Se algum dos números dessa String for negativo, esse método deve lançar uma exceção checked chamada NumeroNegativoException (crie a classe da exceção). Nesse caso, você deve passar para o construtor desta classe uma String que informa o primeiro número negativo passado, caso haja algum. Se todos forem positivos, o método deve retornar a média dos números. Crie a classe e o método de forma que tudo compile. Crie uma nova classe que contenha o método main e que receba um número do console a partir da seguinte mensagem: "Insira uma String que contenha números separados pelo caractere '#':". Caso o usuário digite apenas números positivos na String, deve ser apresentada a mensagem: "A média é: [xxx]", em que [xxx] é a média de todos os números. Caso a exceção seja lançada, imprima no console: "O número [yyy] é negativo, insira apenas números positivos.", em que [yyy] é o número passado no construtor da exceção lançada.