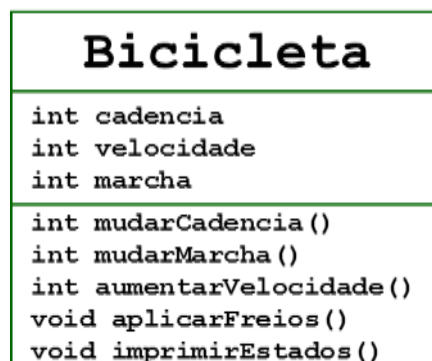


Lista de exercícios 6 LPOO

1. Implemente e entenda os exemplos da aula de hoje;
2. Pesquise na API do Java a classe Integer e responda às questões:
 - a. Qual é a superclasse desta classe?
 - b. Ele implementa alguma interface? Qual? O que ela define?
 - c. Quais outras classes são filhas da superclasse imediatamente superior de Integer?
 - d. Dê pelo menos 3 exemplos de classes que implementam a mesma interface de Integer.
 - e. Existe na classe sobrecarga de construtores ou métodos? Justifique.
 - f. Verifique quais são os métodos estáticos desta classe e dê pelos menos dois exemplos de como seriam utilizados.
 - g. Quais métodos deste objeto são herdados de superclasses?
 - h. Dê um exemplo de upcasting e downcasting utilizando a classe Integer.
3. **(Tarefa 6)** Crie e implemente uma hierarquia de classes para a classe Funcionario. Imagine que Funcionario represente uma classe em um sistema de uma Empresa. Crie classes para herdar de funcionário (Gerente, Diretor e Analista). O gerente deve conter um array de Funcionarios e o Diretor deve conter um array de Departamentos. Implemente um método abstrado na classe Funcionario chamado getBonus. O cálculo do bônus anual segue as seguintes regras. Implemente estas regras utilizando conceitos de polimorfismo:
 - a. Para Diretor: 4 salários + R\$3000,00 por departamento gerenciado
 - b. Para Gerente: 2 salários + R\$100,00 por pessoa gerenciada
 - c. Analista: 1 salário
 - d. Programador: 0,8 salário
 - e. Auxiliar de Limpeza: 0,5 salárioCrie uma classe chamada SistemaRH que instancie os objetos e imprima um relatório com os funcionários da empresa. A classe SistemaRH deve ter um método estático para imprimir o relatório: `public static void imprimeRelatorio(Funcionario[] listaFuncionarios)`
4. **(Tarefa 6)** Modifique o sistema acadêmico criado no item 3 da lista 4 para utilizar orientação a objetos com herança. Crie uma classe pessoa, coloque os atributos da classe aluno que dizem respeito a pessoa nessa classe e faça uma relação de herança entre as classes aluno e professor. Crie a classe professor (deve herdar de pessoa). No sistema acadêmico permita criar professores e relacionar as disciplinas ministradas. Crie uma classe que represente as disciplinas e faça o relacionamento com as classes Aluno e Professor.
5. Declare a herança para a classe Bicicleta, conforme o diagrama do slide 5.
 - a. Procure pensar em estados e comportamentos específicos

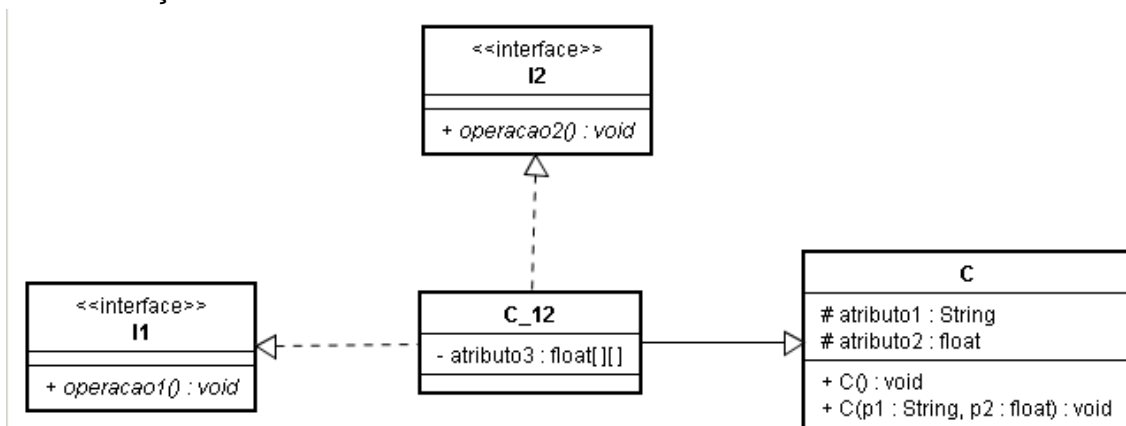


6. Implemente uma classe que utilize as sub-classes da classe bicicleta

7. Procure na API do Java (<http://download.oracle.com/javase/6/docs/api/>) a classe Object
- Estude os métodos clone(), equals(), toString(), hashCode(). Descreva o que eles fazem.
8. Analise o trecho de código abaixo

```
Object a = new Object();
Object b = new Object();
Object c = b;
System.out.println("a==b "+a.equals(b));
System.out.println("b==a "+b.equals(a));
System.out.println("c==b "+c.equals(b));
System.out.println("b==c "+b.equals(c));
System.out.println("a==c "+a.equals(c));
System.out.println("c==a "+c.equals(a));
System.out.println("hashcode a="+a.hashCode());
System.out.println("hashcode b="+b.hashCode());
System.out.println("hashcode c="+c.hashCode());
```

- O que será impresso na saída?
9. Defina a classe Pessoa ou Bicicleta como abstrata.
- Inclua métodos sem implementação.
 - Implemente os métodos abstratos nas subclasses
 - Implemente uma classe para testar a classe Pessoa ou Bicicleta
10. **(Tarefa 6)** Implemente a estrutura de classes representada na figura a seguir, de acordo com as instruções abaixo:



- A implementação de operacao1() deve mostrar no console que passou por tal método e o valor dos atributos do objeto;
- A implementação de operacao2() deve mostrar no console que passou por tal método;
- O construtor C() deve inicializar seus atributos 1 e 2 com "VAZIO" e -999.99f, respectivamente;
- O construtor C(String p1, float p2) deve inicializar seus atributos com os valores providos por parâmetro;
- Os construtores de C_12 devem chamar os construtores respectivos da superclasse e inicializar a matriz atributo3. Se o construtor for sem parâmetros, a matriz deve ser 2 x 2 e seus elementos possuir valor 0. Se o construtor possuir parâmetros, a matriz deve ser 3 x 3 e seus elementos devem possuir valor 1;
- O método main deve possuir o código indicado a seguir. Entenda a origem de todas as saídas;

```
public static void main (String args[]){
    System.out.println ("Iniciando o programa...");
    System.out.println ("Instanciando C1_2 sem parametros:");
    C1_2 objC1 = new C1_2();
    System.out.println ("Instanciando C1_2 com parametros:");
    C1_2 objC2 = new C1_2("XPTO", 3.141516f);
    System.out.println ("Chamando operacoes na instancia 1:");
```

```

objC1.operacao1();
objC1.operacao2();
System.out.println ("Chamando operacoes na instancia 2:");
objC2.operacao1();
objC2.operacao2();
System.out.println ("Instanciando objeto em referencia para interface:");
I1 objC3 = new C1_2("YSBC", 1.99f);
System.out.println ("Chamando operacoes na referencia para interface:");
objC3.operacao1();
}

```

11. **(Tarefa 6)** Altere as classes Quadrado e Circunferencia dos exercícios de aulas anteriores para que implementem a interface Superficie indicada abaixo. Para testar, atribua instâncias de Quadrado e Circunferencia para uma referência à interface e invoque métodos a partir dela.

```

public interface Superficie {
    public double area();
}

```

12. Utilizando a classe Pessoa já criada em exercícios anteriores como superclasse, defina e implemente as subclasses Funcionário (com atributos INumMatricula, iRamal, sCargo e fSalario, getters e setters) e Voluntário (com atributos sHorario e sAtividadePrincipal, getters e setters). Teste a instanciização e utilização dos objetos criados em um programa principal que leia os dados de um funcionário, de um voluntário e a seguir, mostre-os (a leitura deve incluir atributos da classe Pessoa). Mostre também a idade do funcionário e do voluntário;
13. **(Tarefa 6)** Crie uma estrutura de classes que represente diferentes tipos de Veículos. A superclasse chama-se Veiculo. Esta possui três subclasses, VeiculoTerrestre, VeiculoAereo, VeiculoAquatico, que possuem, respectivamente, as subclasses Carro e Trem, Avião e Barco. Crie atributos, getters e setters coerentes com a hierarquia. Toda a hierarquia deve possuir uma operação andar(), que deve ser abstrata nos dois primeiros níveis e implementada nas classes de nível mais baixo. Sua implementação consiste em imprimir no console o movimento executado.
- A classe Veiculo deve possuir um construtor com parâmetros que inicializa seus atributos. As classes filhas devem possuir, também, construtores com parâmetros que, além de chamar o construtor da superclasse, define seus próprios atributos.
 - Para testar o funcionamento das classes, faça um programa que permita ao usuário escolher qual tipo de veículo deseja criar e realize o movimento correspondente. Independentemente do tipo de veículo escolhido, a instância deve ser do tipo Veiculo;
 - Utilizando a hierarquia de classes, faça um programa em que o usuário informa dados de veículos. O programa inicia solicitando ao usuário quantos veículos deseja informar. Para cada um dos veículos, o programa deve identificar seu tipo e ler as informações correspondentes. Os dados informados devem ser armazenados em um único vetor, que deve conter todos os objetos de veículos.
 - Depois de ler todas as informações, o programa deve invocar o método andar() de todos os veículos armazenados.
14. **(Tarefa 6)** Crie uma classe que contenha um método capaz de efetuar a soma das áreas de elementos de um único vetor contendo objetos do tipo Retangulo, Triangulo e Circunferencia. Construa um programa para testar este método.
15. Modifique o código do slide 16 para que todas as saídas sejam impressas com false, mas os objetos b e c devem ser uma cópia um do outro.

- a. Implementar com o método clone.
- b. Dica: Deve ser criada uma outra classe para ser usada no lugar da classe Object que implemente a interface Cloneable

16. Implementar os exercícios das páginas 82, 83 e 84 da apostila da Caelum

- a. <http://www.caelum.com.br/download/caelum-java-objetos-fj11.pdf>