Compartilhar o seu link com: luciorocha @ professores.utfpr.edu.br

Nome: Link:

João Pedro de Paula:< Cópia de POCO4A - Aula 09 - Exercícios >
Isabella Melo Almeida: Cópia de POCO4A - Aula 09 - Exercícios
Deivid da Silva Galvão < Cópia de POCO4A - Aula 09 - Exercícios >
Séfora e Carlos < POO - Aula 09 >
Guilherme Conceição Ramalho < Cópia de POCO4A - Aula 09 - Exercícios >
Matheus Hirata e Thiago Cristovão<> Cópia de POCO4A - Aula 09 - Exercícios
Henrique Cois < Cópia de POCO4A - Aula 09 - Exercícios >
Raphael Uematsu Cópia de POCO4A - Aula 09 - Exercícios
Filipe Augusto Parreira Almeida Cópia de POCO4A - Aula 09 - Exercícios
Luis Henrique Mendes < Cópia de POCO4A - Aula 09 - Exercícios >
João Pedro cavani Meireles<https://docs.google.com/document/d/1ihnkIk2bxEX8BWJSsAwgdCmWGZvq8kfmhayULvk9DWA/edit>
Rafael Kendy Naramoto Lopes < Cópia de POCO4A - Aula 09 - Exercícios >
João Pedro da Silva Kawano < Cópia de POCO4A - Aula 09 - Exercícios >
Gabriel Takeshi < Cópia de POCO4A - Aula 09 - Exercícios >
Vitor Hugo Leite A. de Oliveira < Cópia de POCO4A - Aula 09 - Exercícios >
Marcos Tadao Shoji< Cópia de POCO4A - Aula 09 - Exercícios >
Rodrigo Leandro Benedito: Cópia de POCO4A - Aula 09 - Exercícios
Fernando Rafael: Aula 09 - Exercícios
Thales Alves < POCO4A - Aula 09 - Exercícios >
Angélica B. G. Luciano < Cópia de Aula 09 - Exercícios >
Felipe Antonio Magro< Cópia de POCO4A - Aula 09 - Exercícios >
Daniel Martins de Carvalho < Cópia de POCO4A - Aula 09 - Exercícios >
Victor Ramos Bernardes< Cópia de POCO4A - Aula 09 - Exercícios >
Lucas dos Reis Viana< Cópia de POCO4A - Aula 09 - Exercícios >
Mabylly < Cópia de POCO4A - Aula 09 - Exercícios
Alexandre Aparecido da Silva < Cópia de POCO4A - Aula 09 - Exercícios >
Plinio < Cópia de POCO4A - Aula 09 - Exercícios >

Revisão de Conteúdos

Quizz 2: https://quizizz.com

1) Marque Verdadeiro ou Falso:

a(V) O construtor é a primeira instrução executada quando um objeto é instanciado.
b(V) O método super invoca o construtor da superclasse.
c(V) A composição é um relacionamento do tipo “tem um” entre as classes.
d(V) A herança é um relacionamento do tipo “é um” entre as classes.
e(V) A composição e a herança podem existir no mesmo código.
f(V) A classe é um modelo a partir do qual objetos são instanciados.
g(V) O objeto é uma instância da classe.
h(V) Os membros da classe são apenas os atributos e os comportamentos.
i(V) Os membros static são compartilhados entre todos os objetos.
j(V) Os membros privados são ocultados para outras classes.

2) Analise o código-fonte a seguir:

```
/*
                                    Empresa
                                    |       |
                          Floricultura      Fornecedor

*/
public class Empresa {

}
public class Floricultura extends Empresa {

}
public class Fornecedor extends Empresa {

}
public class Sistema {

  public static void main( String [ ] args) {
          new Sistema().iniciar();
  }
  public void iniciar() {
       Fornecedor fornecedor = new Fornecedor();
  }
}
```

Marque Verdadeiro ou Falso:

(V) As classes folha são Floricultura e Fornecedor.

(V) A única superclasse descrita é a classe Empresa.
(V) Um atributo protected em Empresa estará imediatamente disponível para as classes Floricultura e Fornecedor.
(V) Um método private em Empresa não estará disponível para as classes Floricultura e Fornecedor.
(V) Um método sem modificador de acesso estará imediatamente disponível para as classes Floricultura e Fornecedor.

3) Observe o diagrama UML de classes da Figura 3:

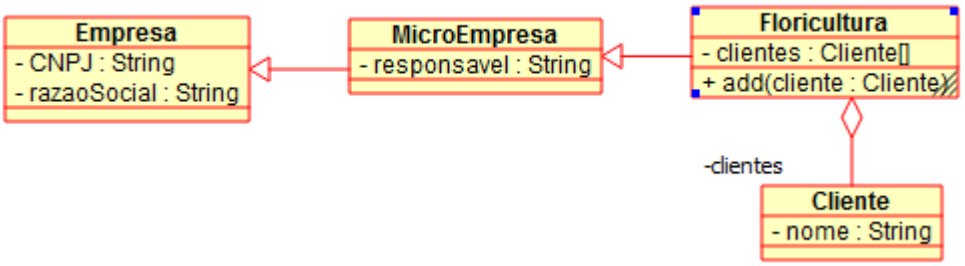


Figura 3 - Diagrama UML de Classes.

a) Preencha a tabela a seguir, conforme descrito na Figura 3.

Classe	Subclasse	Superclasse
Empresa	MicroEmpresa	—-
MicroEmpresa	Floricultura	Empresa
Floricultura	—--	MicroEmpresa
Cliente	—--	—--

b) Apresente a modelagem da Figura 3 em linguagem orientada a objetos.

```
public class Empresa {
  private String CNPJ;
  private String razaoSocial;
}
public class MicroEmpresa extends Empresa {
  private String responsavel;
}
public class Floricultura extends MicroEmpresa {
    private ArrayList<Cliente> lista;
    public void add(Cliente cliente){
       //TODO
    }
}
```

```
public class Cliente {
   private String nome;
}
```

4) Descrição: Um sistema de correios recebe cartas. Cada carta tem um selo. A carta pode ser convencional ou expressa.

a) Preencha a tabela conforme a descrição:

Classe	Variáveis de instância	Métodos
SistemaDeCorreios	ArrayList<Carta> lista	getCartas
Cartas	String tipo (convencional ou expressa) Selo selo	
Selo		

b) Apresente a modelagem em linguagem orientada a objetos.

```
public class SistemaDeCorreios {

        ArrayList<Carta> lista;

        public  ArrayList<Carta> getCartas(){
                return lista;
        }
}
public class Cartas {
      private String tipo;
      Selo selo;
}
public class Selo {

}
```

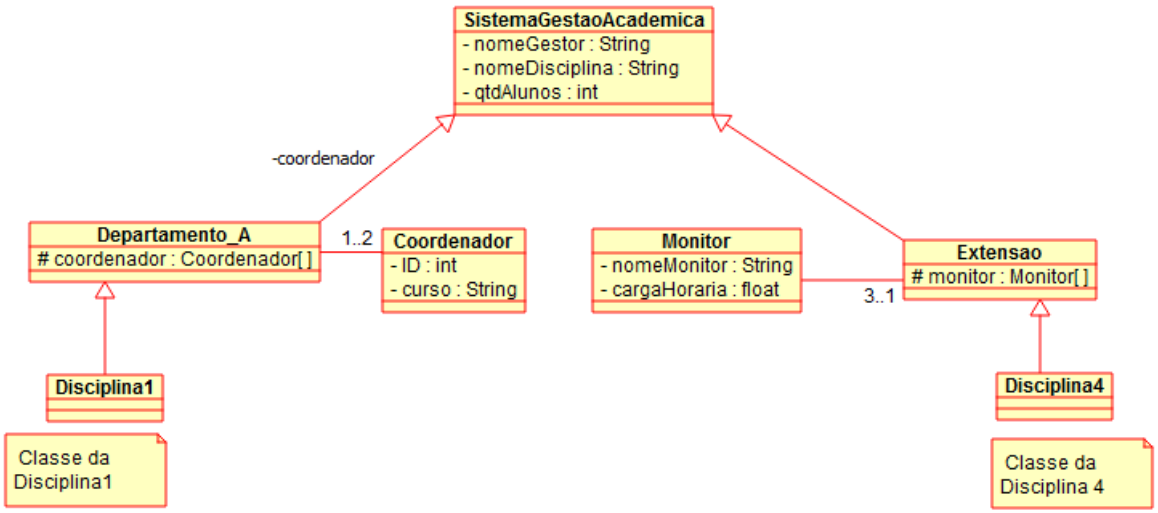
5) Descrição: Um sistema de correios é um sistema de encomendas. Um sistema de encomendas é um software comercial. O sistema de correios deve cadastrar o remetente da encomenda.

a) Preencha a tabela conforme a descrição:

Classe	Variáveis de instância	Métodos

b) Apresente a modelagem em linguagem orientada a objetos.

6) Dado o diagrama UML a seguir, preencha a tabela a seguir, com os construtores.



Classe	Construtor
Disciplina1	public Disciplina1()
Disciplina1: Chamada da Superclasse	super(nomeGestor, nomeDisciplina, qtdAlunos, coordenador);
Departamento_A	public DepartamentoA(String nomeGestor, String nomeDisciplina, int qtdAlunos, ArrayList<Coordenador> coordenador)
Departamento_A: Chamada da superclasse	super(nomeGestor, nomeDisciplina, qtdAlunos)
SistemaGestaoAcademica	public SistemaGestaoAcademica(String nomeGestor, String nomeDisciplina, int qtdAlunos)
SistemaGestaoAcademica: Chamada da superclasse	—------
Coordenador	public Coordenador(int ID, String curso)
Coordenador: Chamada da superclasse	—-----

Monitor	public Monitor(String nomeMonitor, float cargaHoraria)
Monitor: Chamada da superclasse	—------

7) Preencha a tabela a seguir:

Modificador de acesso	Classe	Package	Subclasse	Outro Package
public	TRUE	TRUE	TRUE	TRUE
Sem modificador	TRUE	TRUE	FALSE	FALSE
protected	TRUE	TRUE	TRUE	FALSE
private	TRUE	FALSE	FALSE	FALSE