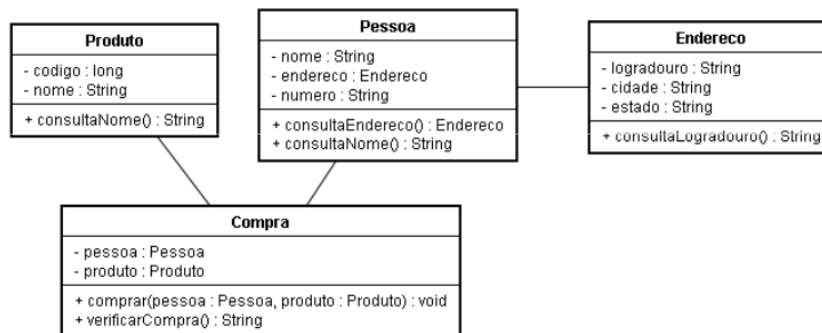
 FASAM FACULDADE SUL-AMERICANA	DATA:
Lista POO 2	
CURSO: Sistemas de Informação	NOTA:
DISCIPLINA: Programação Orientada a Objetos	
PROFESSOR: Luiz Mário Lustosa Pascoal	
ALUNO:	

1. Implemente em Java um programa para controle da situação dos alunos de Sistemas de Informação na disciplina de Programação Orientada a Objetos. Crie uma classe **Aluno** com os seguintes atributos: *n° de matricula*, *nome*, *notaBimestre1*, *notaBimestre2*, em seguida insira o construtor (que deverá conter estes 4 atributos) e métodos Get e Set para os mesmos. Em seguida, crie a classe **Principal** que deverá conter o método *main* e será responsável por ler os 4 atributos de pelo menos 3 alunos utilizando o Scanner ou JOptionPane, e para cada leitura completa deverá criar um objeto do tipo Aluno, onde cada objeto Aluno criado deverá ser armazenados em um ArrayList <Aluno>, que representa uma lista de alunos. Por fim, percorra a lista de alunos informando a situação dos alunos presentes na listaAlunos, caso a média aritmética entre *notaBimestre1* e *notaBimestre2* seja maior ou igual a 6.0, o programa deverá informar que o aluno está APROVADO, caso contrário informe REPROVADO.
2. No pacote entidade crie as seguintes classes : **Ator**, **Diretor** e **Filme**. Para a classe **Ator** defina os seguintes atributos: **nome (String)**, **idade (int)**, **n_oscars (int)** (número de Oscars conquistados). Para a classe **Diretor** defina os atributos: **nome (String)**, **ganhador_oscar (boolean)**, **principal_filme (Filme)**. Para a classe **Filme** defina o atributo **título (String)** e as referências : **diretor (Diretor)**, **ator_principal (Ator)**, **atriz_principal (Ator)**. Crie um construtor na classe **Diretor** para deixar a inicialização de **principal_filme** para ser realizada através do método **setPrincipalFilme**. No pacote controle crie a classe **CadastroFilme** para criar um diretor, um ator, uma atriz e um filme, e imprimir, em cada linha, a seguinte descrição para o filme: título do filme, nome do diretor, nome do ator principal, nome da atriz principal. Imprima, adicionalmente, uma descrição para o diretor contendo em cada linha: nome do diretor, nome do principal filme. Os dados das classes deverão ser privados e os métodos não auxiliares deverão ser públicos.
 - a. Altere a classe CadastroFilme do exercício anterior para criar objetos e armazenar em arrays estáticos de cada uma das classes entidades os objetos criados. Crie 4 filmes com alguns diretores e atores repetidos. Defina o método **Filme buscaFilme (String título)** e o método **boolean insereFilme (Filme)** que retorna false se já existir um filme com o título do novo filme. A classe CadastroFilme deverá verificar se o título do filme já foi cadastrado antes de solicitar a inserção do novo filme. Informe sucesso ou insucesso em cada inserção.
 - b. Por fim, defina também um método para buscar diretores vencedores de oscars. Percorra o vetor de diretores e o método deverá imprimir na tela o **nome do diretor, o nome do filme, o nome ator e da atriz principal do principal filme** ganhador do oscar.

3. Utilize a implementação de **Pessoa** e **Endereço** feita em sala de aula e adicione as seguintes classes ao relacionamento:



A classe **Compra** vai conter um objeto do tipo **Pessoa** e outro do tipo **Produto**. É possível comprar, passando como parâmetro dois objetos do tipo Pessoa e outro do tipo Produto.

Em seguida, desenvolva a classe Principal da seguinte forma:

- Inicialmente, a classe deve criar 2 objetos do tipo Produto (crie os objetos com as informações que desejar)
 - O usuário faz um cadastro (criando um objeto do tipo Pessoa)
 - Logo após, o usuário seleciona entre os 2 produtos cadastrados anteriormente
 - Dependendo da escolha, é acessado o método comprar da classe Compra, e passado como parâmetro o objeto Pessoa que ele cadastrou e o do Produto escolhido
 - Logo após, é exibido no console ao usuário uma mensagem de confirmação, exibindo o nome dele e do produto comprado (acesso ao método verificarCompra()) e finaliza a aplicação
4. Utilize a implementação das classes Pessoa e Dado feitas em sala de aula e altere o programa de acordo com o que se pede a seguir:

Crie a classe Endereco

- Esta classe possui o nome da rua, um número na rua, uma cidade, um estado, um CEP e um bairro
- Quando um endereço é criado, deve ser informado o nome da rua, o número na rua e a cidade
- Crie os métodos de acesso da classe (getters e setters)

Crie a classe Data

- Esta classe possui um dia, um mês e um ano
- Ao criar uma data, todos os atributos devem ser informados (dia, mês e ano)
- Crie os métodos de acesso da classe (getters e setters)

Altere a classe Pessoa, onde:

- Esta classe agora possui um nome, um sexo, uma data de nascimento (do tipo Data), um endereço (do tipo Endereco) e dois dados padrão (do tipo Dado – Dica: olhar slide 5) um de 6 lados (meuDadoDe6faces) e um de 8 lados (meuDadoDe8faces), que devem ser criados no momento da instanciação de uma Pessoa. Pense no método construtor da classe Pessoa, e analise os parâmetros de entrada deste método
- Crie dois métodos para jogar respectivamente os dois dados de 6 e 8 faces. Os métodos jogarDadoDe6Faces() e jogarDadoDe8Faces() deverão utilizar a classe Random do java.util para aleatoriamente gerar um número que representa a face obtida no lançamento do dado.
- Crie os métodos de acesso da classe (getters e setters). Pense nos tipos de retorno dos métodos get e nos parâmetros de entrada dos métodos set

Na classe Principal:

Dois jogadores lançam ao mesmo tempo dois dados, um de 6 faces e outro de 8 faces. O jogador que obtiver a soma das faces dos dois dados, mais alta vence a rodada marcando um ponto. O jogo termina após uma melhor de 7 rodadas. Escreva um programa que simule este jogo de dados. Dica: use os métodos jogarDadoDe6Faces() e jogarDadoDe8Faces() para gerar os números resultantes dos lançamentos dos dados de cada um dos jogadores. A cada rodada de lançamento, o programa deverá mostrar o valor de cada dado jogado por cada um dos 2 jogadores e mostrar quem venceu a rodada. O jogador que obtiver mais rodadas vencidas vence o jogo. Realize este jogo para 6 pessoas diferentes, onde elas se enfrentaram de duas a duas, totalizando assim 3 jogos distintos. Os nomes dos vencedores das três partidas deverão ser armazenados em um Vetor do tipo String, que será utilizado para imprimir o nome dos vencedores ao final do programa.