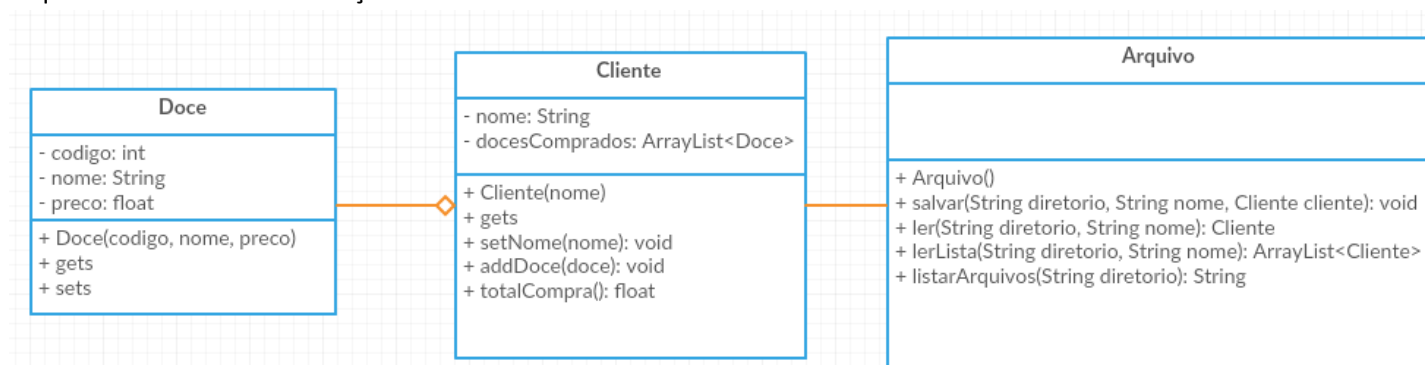


**PRÉ PROVA 2****OBSERVAÇÕES GERAIS:**

- (i) A pré prova pode ser feita em dupla e é com consulta.
- (ii) Questões que apresentarem cópias dos colegas ou da internet serão anuladas
- (iii) Questões que não compilam serão anuladas
- (iv) Deve ser postado no AVA SOMENTE os arquivos .java dos exercícios, podendo estar dentro de seus respectivos pacotes.
- (v) Para cada exercício será avaliado o raciocínio lógico, o uso de boas práticas, capacidade de interpretar e resolver problemas e a capacidade de abstração, coerência e lógica das especificações realizadas.

**(8,0) 1** Implemente o problema seguindo o diagrama de classes seguinte. Os métodos correspondentes ao arquivo texto devem utilizar as classes do pacote *nio* do Java. O texto deve ser gravado de forma legível, sem utilização de arquivos binários ou serialização.



Teste/main:

```

Arquivo arquivo = new Arquivo();
Cliente c1 = new Cliente("Carla");
c1.addDoce(new Doce(1, "Bala de Goma", 1f));
c1.addDoce(new Doce(2, "Bolo de amora", 10f));
arquivo.salvar("Dia 1", "Arquivo01", c1);

Cliente c2 = new Cliente("Josué");
c2.addDoce(new Doce(1, "Bala de Goma", 1f));
arquivo.salvar("Dia 1", "Arquivo01", c2);

Cliente cliente = arquivo.ler("Dia 1", "Arquivo01");
System.out.println(cliente.toString());

ArrayList<Cliente> clientes = arquivo.lerLista("Dia 1", "Arquivo01");
for (Cliente cli: clientes) {
    System.out.println(cli.toString());
}

String arquivos = arquivo.listarDiretorios("Dia 1");
System.out.println(arquivos);
  
```

Descrição dos métodos:

Classe	Método	Descrição	Pontuação
Doce	gets	Retornar o atributo	0
Doce	sets	Validar e alterar o atributo	0,5
Cliente	gets	Retornar o atributo	0
Cliente	setNome	Validar e alterar o nome	0,5
Cliente	addDoce	Validar e adicionar o doce na lista	0,5
Cliente	totalCompra	Retornar o valor total da compra	0,5
Arquivo	salvar	<p>Deve salvar um arquivo texto no diretório e nome informado no parâmetro da seguinte forma:</p> <p>&lt;Nome do Cliente&gt;            &lt;Codigo Doce&gt;            &lt;Nome Doce&gt;            &lt;Preco Doce&gt;            &lt;Codigo Doce&gt;            &lt;Nome Doce&gt;            &lt;Preco Doce&gt;            &lt;Codigo Doce&gt;            &lt;Nome Doce&gt;            &lt;Preco Doce&gt;            &lt;....&gt;            &lt;Valor total compra&gt;</p> <p>Pode ter mais de um cliente no mesmo arquivo. Nesse caso, o próximo cliente será gravado ao final do arquivo.            Se o diretório especificado não existir, o mesmo deve ser criado na pasta do projeto.            Se o arquivo especificado não existir, o mesmo deve ser criado na pasta do diretório especificado.            Se o arquivo existir, ele não deve ser sobrescrito.            O arquivo deve ser gerado com a extensão “.cli”            O arquivo deve ser legível.            Se qualquer erro ocorrer deve-se informar o motivo do erro através de uma exceção.</p>	2,0
Arquivo	ler: Cliente	<p>Retorna em um objeto Cliente os dados do primeiro cliente que se encontra no arquivo especificado. O arquivo é composto pelo diretório e pelo seu nome, indicados no parâmetro.</p> <p>Se o diretório ou o arquivo não existir, ou se o arquivo estiver em branco, deve-se lançar uma exceção indicando qual foi o problema ocorrido.</p>	1,5
Arquivo	lerLista: ArrayList	<p>Retorna em um ArrayList de Clientes os dados de todos os clientes que se encontram no arquivo especificado. O arquivo é composto pelo diretório e pelo seu nome, indicados no parâmetro.</p> <p>Se o diretório ou o arquivo não existir, ou se o arquivo estiver em branco, deve-se lançar uma exceção indicando qual foi o problema ocorrido.</p>	1,5
Arquivo	listarArquivos	Retornar em uma String separada por quebra de linha o nome de todos os arquivos que estão dentro do diretório especificado.	1,0

**(1,0) 2)** Faça com que seja gravado um objeto serializado para cada doce inserido na lista de compras do cliente. O arquivo deve possuir o nome do cliente, com extensão “.ser” e deve estar na pasta do projeto. Lembre-se que a pasta do projeto pode estar em diretórios diferentes em cada computador. O arquivo deve salvar na pasta do projeto independente do

caminho principal. Deve haver um arquivo por cliente, contendo todos os doces de forma serializada. Deve ser gravado um doce de cada vez, sem a utilização do tipo genérico “Object”.

**(1,0) 3)** Desenhe o diagrama de sequências para a ação de adicionar um doce e salva-lo no arquivo, representando a situação “comprar um doce” feita pelo cliente.