INSTITUTO FEDERAL Bahia – Campus Vitória da Conquista	**AVALIAÇÃO 1**	
	CURSO: Bacharelado em Sistemas de Informação	**MODALIDADE:** Ensino Superior
	MÓDULO/SEMESTRE/SÉRIE: 3º	**PERÍODO LETIVO:** 2024.1
	DISCIPLINA: Linguagem de Programação II	**CLASSE:** 20241.3.119.1N
	DOCENTE: Alexandro dos Santos Silva	
	DISCENTE:	**CONCEITO:**

INSTRUÇÕES

- A avaliação é composta de 5 (cinco) questões com escore total de 7 (sete) pontos;
- **Em caso de identificação de plágio, todos os discentes envolvidos terão sua avaliação anulada, com consequente atribuição de conceito nulo**;
- Para resolução das questões, será admitido o uso apenas da sintaxe adotada para escrita de programas em Java;
- Será admitida entrega da resolução apenas através do ambiente da disciplina junto à plataforma Google Sala de Aula e em prazo estabelecido pelo docente.
- Para efeito de entrega da resolução, **todos os arquivos de extensão .java gerados devem ser compactados em um único arquivo, cujo nome deve corresponder ao próprio nome do discente**, sob pena, inclusive, de penalização.

---

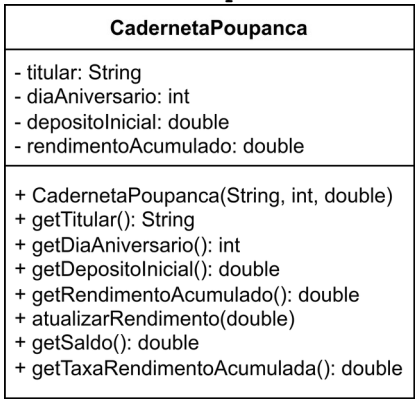
1. (Peso: 1,5) A caderneta de poupança é um tipo de investimento de renda fixa com rentabilidade padronizada definida pelo governo através do Banco Central do Brasil. Na prática, a quantia é depositada em uma conta poupança e, a cada mês, é aplicado um percentual de rendimento sobre aquele valor depositado. Essa remuneração, contudo, é creditada apenas na data de "aniversário", que nada mais é do que o dia do mês em que o depósito foi feito. A título de exemplificação, caso uma pessoa tenha depositado a quantia no dia 22 (vinte e dois) de março, a rentabilidade e o crédito correspondente na conta daquele pessoa ocorrerá somente no dia 22 (vinte e dois) dos meses posteriores. No quadro abaixo, estão listadas taxas de rendimento da poupança nos últimos 6 (seis) meses, conforme dados disponíveis em **http://www.ipeadata.gov.br/ExibeSerie.aspx?serid=31878&Module=M**:

Mês	Taxa de Rendimento (%)
09/2023	0,61
10/2023	0,61
11/2023	0,58
12/2023	0,57
01/2024	0,59
02/2024	0,5

Em caso de abertura de uma caderneta de poupança no valor de R$ 125,00 no dia 22/09/2023 e de modo a não haver nenhum depósito posterior, o saldo atualizado da caderneta no dia 22/03/2024 seria de exatos R$ 129,38 após crédito, mês a mês, das renumerações de acordo com as taxas de rendimento mostradas no quadro acima **(rendimento acumulado no período seria de exatos de R$ 4,38, o que corresponde a uma taxa de rendimento acumulada de aproximadamente 3,50%)**. Posto isto, implemente uma classe para fins de encapsulamento de caderneta de poupança, de nome `CadernetaPoupanca`. Na forma de campos de instância, referidos dados incluem nome do titular, dia de aniversário (um valor entre 1 e 31) e valor de depósito inicial, de modo a serem inicializados na **forma de passagem de parâmetros ao construtor** no momento em que novo objeto da classe for instanciado. Considere ainda inclusão de campo de instância adicional para registro de **rendimento acumulado** da caderneta de poupança. Além dos métodos de acesso *getter* de tais campos de instância, exige-se a inclusão de outros métodos cuja especificação se segue abaixo:

Método	Descrição
`void atualizarRendimento(double prTaxa)`	Atualização de rendimento acumulado considerando-se **saldo atualizado** e taxa percentual de rendimento indicada através do parâmetro `prTaxa`
`double getSaldo()`	Retorno de saldo atualizado da caderneta de poupança considerando-se **depósito inicial** e **rendimento acumulado**
`double getTaxaRendimentoAcumulada()`	Retorno de taxa de rendimento acumulada considerando-se valores de **depósito inicial** e de **rendimento acumulado**

Diagrama de classes contendo definição da classe `CadernetaPoupanca` da forma como sugerido acima segue-se abaixo:



2. (Peso: 1,5) Implemente classe utilitária de nome **`CadernetaPoupancaUtil`** para demonstração das capacidades da classe **`CadernetaPoupanca`** da questão anterior. A referida classe deve dispor de método estático **`main`** de modo a permitir que sejam implementadas operações de entrada, pelo usuário, dos dados de 7 (sete) cadernetas de poupança. Sobre estes dados, são eles: nome do titular, dia de aniversário e valor de depósito inicial (a serem encapsulados em *array* de objetos da classe **`CadernetaPoupanca`**). Após isso, deverá ser executada alguma das seguintes operações:
   a) Atualização de rendimento acumulado de alguma das 7 (sete) cadernetas de poupança, cabendo ao usuário, para tal, informar nome do titular e taxa percentual de rendimento (deverá ser exibido saldo da caderneta antes e após aplicação de taxa percentual de rendimento);
   b) Listagem de nomes de titulares e saldos de cadernetas de poupança de determinado dia de aniversário considerando-se que caberá ao usuário previamente informar dia desejado para que sejam selecionadas apenas cadernetas com aniversário naquele dia informado;
   c) Listagem de depósito inicial, dia de aniversário e saldo de cadernetas de poupança de determinado titular considerando-se que caberá ao usuário previamente informar nome do titular para que sejam selecionadas apenas cadernetas em nome daquele titular (para localização de referências de instâncias de cadernetas de poupança armazenadas no *array*, caberá invocação do método **`equalsIgnoreCase`** da classe **`java.lang.String`**);
   d) Encerramento da aplicação.

   Observação: em caso de seleção de alguma das operações elencadas pelos itens *a*, *b* e *c*, após executada, deverá ser permitido ao usuário selecionar novamente outra operação.

3. (Peso: 1,5) Readeque as classes das 2 (duas) questões anteriores de acordo com condições que se seguem abaixo:
   a) Redefinição de construtor da classe **`CadernetaPoupanca`** considerando lançamento de exceção da classe **`java.lang.Exception`** em caso de inicialização de campo de instância de dia de aniversário com qualquer valor abaixo de 1 (um) ou superior a 31 (trinta e um) e/ou inicialização de campo de instância de depósito inicial com valor igual ou abaixo de 0 (zero);
   b) Redefinição de método **`atualizarRendimento`** da classe **`CadernetaPoupanca`** considerando lançamento de exceção da classe **`java.lang.Exception`** em caso de tentativa de aplicação de taxa percentual de rendimento igual ou inferior a 0 (zero);
   c) Inclusão, em método estático **`main`** da classe utilitária **`CadernetaPoupancaUtil`**, de blocos **`try/catch`** para captura e tratamento de eventuais exceções lançadas ao se invocarem construtor ou método **`atualizarRendimento`** da classe **`CadernetaPoupanca`** (o tratamento consistirá na exigência de entrada, novamente, dos dados da caderneta ou da taxa percentual de rendimento até que não haja mais lançamento de exceção).

4. (Peso: 1,0) Readeque a classe **`CadernetaPoupanca`** da questão anterior considerando que ela estenda a classe **`java.lang.Thread`** e de modo, por consequência, a incluir sobrescrita do método **`run`**. A sobrescrita de tal método consistirá na atualização **contínua**, a cada 2 (dois) segundos, de rendimento acumulado da caderneta de poupança considerando-se taxa percentual de rendimento obtida de forma pseudoaleatória entre 0,5 (inclusive) e 1,0 (exclusive). Para a geração de números pseudoaleatórios, sugere-se o uso de método estático **`nextDouble`** da classe **`java.util.Random`** conforme se segue abaixo:

```
01  // Gerador aleatório de números pseudoaleatórios
02  java.util.Random gerador = new java.util.Random();
03  // geração de número de ponto flutuante pseudoaleatório entre 0.5 (inclusive) e 1 (exclusice)
04  double numero = gerador.nextDouble(0.5, 1);
```

5. (Peso: 1,5) Para demonstração das capacidades da classe readequada na questão anterior, implemente nova classe utilitária, de nome **`CadernetaPoupancaThreads`**. Referida classe utilitária deve dispor de método estático **`main`** através do qual sejam executadas três *threads* baseadas na readequação da implementação da classe **`CadernetaPoupanca`** da forma como exigido na questão anterior. No tocante à instanciação das instâncias da classe **`CadernetaPoupanca`**, caberá ao usuário informar um único nome de titular e dia de aniversário; elas se distinguirão entre si apenas em relação ao valor de depósito inicial (a ser igualmente informado pelo usuário para cada uma das cadernetas de poupança). Após isso, um bloco de repetição deve ser implementado para permitir ao usuário, a qualquer instante, consultar saldo atualizado de alguma das 3 (três) cadernetas de poupança ou, ao invés disso, encerrar o programa em definitivo. Quando da realização da consulta, caberá ao usuário informar de qual das cadernetas de poupança (1, 2 ou 3) deseja consultar o saldo atualizado.