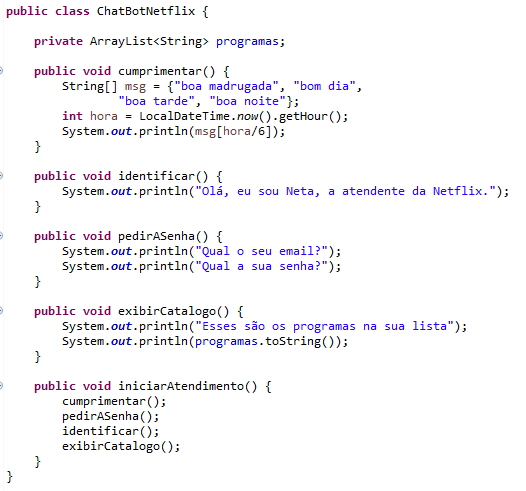
ADS – IFPB – Campus Monteiro

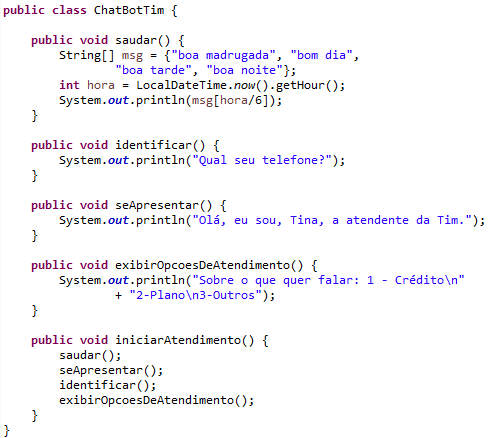
LISTA SEMANAL – PROGRAMAÇÃO II - PROGRAMAÇÃO ORIENTADA A OBJETOS  
Prof. Cleyton Caetano de Souza

**Observação:** Essa lista pode ser feita em dupla.

Semana 9

1. Pesquise[[1]](#footnote-1) e escreva uma redação enumerando as principais vantagens e desvantagens do uso de herança. Sua redação tem de ter, ao menos, 10 linhas (fonte Arial 11), e precisa usar o conteúdo de, pelo menos, 3 fontes diferentes (incluir as fontes). Você pode usar como fontes: livros, vídeos no youtube, artigos de sites, respostas em sites de perguntas e respostas e etc. (essa questão pode ser entregue digitada - **sem copiar e colar, por favor**).

2. Imagine que você foi contratado por uma empresa que desenvolve *chatbots*. Um *chatbot* é um programa de computador que tenta simular um ser humano na conversação com as pessoas. Atualmente, muitas empresas vêm usando *chatbots* para intermediar a comunicação entre o cliente e a atendente humana. Ao analisar o código de dois *chatbots* para duas empresas diferentes, você se deparou com as seguintes classes:

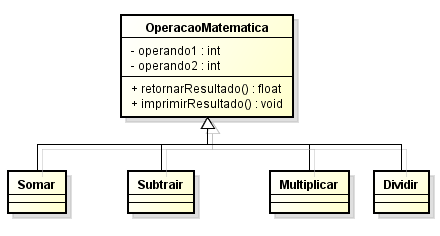


Após analisar o código dos *chatbots*, você identificou várias semelhanças entre as duas classes e uma oportunidade de reduzir a duplicação de código aplicando um padrão de projeto chamado *template method* (você é muito perspicaz, não é?)*.* Reescreva o código dos *chatbots* aplicando o padrão de projeto e explique porque essa é uma solução melhor para o design dessas classes (você tem liberdade para mudar quaisquer aspectos das classes envolvidas).

3. Organize as seguintes classes numa árvore de herança: Avião, Carro de Fórmula 1, Ferrari F430, Ferry, Barco de Cruzeiro, Submarino Nuclear, Submarino não pilotado, Veículo, Veículo Terrestre, Veículo Aquático, Ferrari, Barco a Remos, F16, Airbus A380, Boeing757, Balão, Carro Desportivo, Veleiro, Veículo Aéreo, Espaço nave (você pode incluir outras classes que irão facilitar a elaboração dessa árvore de herança). Sinalize (usando <abstract> acima do nome da classe) quais as classes que potencialmente serão abstratas.

4. Crie uma classe para representar uma Conta Corrente, com métodos para depositar uma quantia, sacar uma quantia e obter o saldo. **O setSaldo deve ser protected**. Adicione os atributos titular, número da conta e saldo. A cada saque realizado deve ser debitada uma taxa de operação equivalente a um real. Crie, em seguida, uma subclasse desta classe anterior para representar uma conta corrente especial (para clientes especiais). Clientes especiais pagam taxas de operação de apenas 50 centavos (isso quer dizer que sempre que sacar da ContaEspecial for chamado, deve-se debitar 50 centavos do saldo). Crie um programa que permita ao usuário escolher entre criar uma classe conta corrente normal ou especial, permita a ele definir um valor inicial para saldo da conta, em seguida, peça para que ele saque um valor e informe a ele o valor do novo saldo.

5. Codifique a seguinte árvore de herança.



O método retornar resultado deve retornar o resultado da operação matemática, enquanto o método imprimir resultado deve imprimir o resultado no console (identifique onde esses métodos devem ser implementados e quais classes devem ser concretas e quais devem ser abstratas). Ao final, crie um **programa** que permita que o usuário escolha qual conta ele deseja fazer (escrevendo a operação por extenso: “adição”, “subtração”, “multiplicação”, “divisão”), e quais os parâmetros de entrada. Instancie e use os subtipos de OperacaoMatematica para atender à solicitação do usuário e informar o resultado da operação.

6. Observe o programa desenvolvido no programa da questão 4 e 5 e responda: você acha que há código duplicado nele? Por quê? (essa questão pode ser entregue digitada - **sem copiar e colar, por favor**).

7. Pesquise e descreva, com suas palavras, o que você entende por polimorfismo. Você consegue ver alguma possibilidade de usar o polimorfismo para evitar o código duplicado nas questões 4 e 5? (essa questão pode ser entregue digitada - **sem copiar e colar, por favor**).

1. Embora o que você já sabe seja muito importante, eu não quero que você escreva apenas com base na sua cabeça. Eu quero que você leia artigos e outros tipos de textos de outros programadores, antes de escrever esse texto. Se quiser, você pode até perguntar em algum Site de Perguntas e Respostas (como Stack Overflow ou GUJ, por exemplo). [↑](#footnote-ref-1)