



## Grande Área 2 Aula Prática POO

**Professores (as):**

Alex Andrade

**Curso:** \_\_\_\_\_

**Data:** \_\_\_\_/\_\_\_\_/\_\_\_\_

### Dados da Equipe:

Aluno (a): _____	RA: _____
Aluno (a): _____	RA: _____
Aluno (a): _____	RA: _____
Aluno (a): _____	RA: _____
Aluno (a): _____	RA: _____

### Instruções

1. Essa atividade será desenvolvida em grupo de até 5 participantes.
2. Os arquivos de desenvolvimento do projeto DEVEM SER COLOCADOS EM UMA ÚNICA PASTA e, em seguida, a pasta deve ser ZIPADA/COMPACTADA e enviada para os respectivos professores.

**Para esta atividade, considerem o seguinte cenário:**

- A empresa **SegPOO**, especializada em alugueis de meios de transportes terrestres, contratou a sua equipe para criar um programa que fosse possível realizar cadastros dos seus automóveis. Ao conversar com o representante da empresa, ele informou que são alugados:  
**caminhões, tratores, carros de passeio, motos e bicicletas.**
- Desse modo, sua equipe deve criar uma programação que possua uma **classe super** chamada de TRANSPORTES e 5 subclasses que devem herdar atributos da mesma (HERANÇA).
- Para a sua superclasse deve ser criado 5 atributos e para cada subclasse, além de herdar os atributos da super, devem ter, **no mínimo**, mais 3 atributos correspondentes (ABSTRAÇÃO).
  - Obrigatoriamente, um dos atributos da SUPERCLASSE será um NÚMERO DE CARTÃO DE CRÉDITO, que será fornecido na hora do aluguel do devido meio de transporte.
    - Deve ser criado um método que valida o número do cartão de crédito antes de atribuí-lo ao atributo (algoritmo de um validador de cartão está disponibilizado no fim desse arquivo).
- Os atributos de todas as classes devem estar encapsulados (ENCAPSULAMENTO). Logo, devem ser acessados através dos métodos GET's e SET's.
- Deverá ser implementado pelo menos 3 métodos, na classe TRANSPORTES, que serão herdados para as subclasses e UM MÉTODO, pelo menos, deverá ter sua lógica modificada nas subclasses (POLIMORFISMO).
- Na classe principal do programa, deve-se cadastrar 4 transportes de cada tipo (subclasses). Em seguida deve ser impresso uma mensagem de boas-vindas ao programa e TODOS OS ATRIBUTOS correspondentes aos transportes instanciados.

### Algoritmo validador de número de cartão de crédito:

Usuário entra com os 16 dígitos, referente ao cartão de crédito/débito do cliente, e o algoritmo retorna se ele é válido, além da Bandeira do Cartão.

#### Estrutura do Algoritmo:

Números de um cartão:

- Os 6 primeiros dígitos/números referem-se à bandeira do cartão.
  - Os 9 dígitos seguintes identificam o cliente;
  - O 16º dígito é o verificador.
- Os cartões de crédito/débito só podem iniciar com 3 ou 4 ou 5 ou 6.
  - Bandeiras a se verificar:
    1. Visa - Então a sequência começa com 4;
    2. Mastercard - Será um número entre 51 e 55;
    3. American Express - Será 34 ou 37;
    4. Maestro - Será 50 ou um número entre 56 e 69.
    5. Especificar como "Outra Bandeira" as demais entradas que podem ser válidas.

O cálculo do dígito verificador é:

1. Multiplica o primeiro, terceiro, quinto, ..., décimo quinto dígito por 2.
2. Se isso resultar em algum número com dois dígitos, você soma os dois (por exemplo, 18 vira 1+8=9). **DICA: TEM DE CONVERTER PARA STRING E DEPOIS PEGAR AS POSIÇÕES DA STRING E SOMA-LAS, CONVERTENDO ASSIM PARA INTEIRO NOVAMENTE.**
3. Depois some tudo.
4. Adicione o segundo, quarto, ..., décimo quarto dígito.
5. Suponha que o resultado deu 67. Quanto falta para chegar a um múltiplo de dez? 3.
6. Então o dígito verificador tem de ser igual a 3.