
	<div>  <div> <b>INSTITUTO FEDERAL</b>            Paraíba            Campus João Pessoa         </div> </div> <div> <b>Programação e Estrutura de Dados</b>  <b>Professor:</b> Alex Sandro da Cunha Rêgo  <b>Última atualização:</b> 10/10/2018         </div>	<div> <b>Lista</b>  <b>6</b> </div>
<b>CLASSES E OBJETOS</b>		

Pré-Requisitos
<ul style="list-style-type: none"> <li>Lógica de programação, conhecimento na criação de classes e objetos em Python: construtor, métodos e propriedades públicas e privadas, métodos e propriedades de classe e de instância.</li> </ul>

- 1) Codifique uma classe em que cada objeto representa um voo que acontece em determinada data e em determinado horário. Cada voo possui no máximo 100 passageiros e a classe deve controlar a ocupação das vagas. A classe deve dispor dos seguintes métodos:

<b>Construtor()</b>	Configura os dados do voo (recebidos como parâmetro): número do voo, data (para armazenar a data utilize um objeto da classe Data, criada na Lista 05);
<b>getAssento()</b>	Retorna o número da próxima cadeira livre
<b>estaDisponivel(numCadeira)</b>	Verifica se o número da cadeira recebido como argumento está livre ou não
<b>ocupar(numCadeira)</b>	ocupa determinada cadeira do voo, cujo número é recebido como parâmetro. Retorna verdadeiro se a cadeira ainda não estiver ocupada (operação foi bem sucedida) e falso caso contrário
<b>vagasDisponiveis()</b>	retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo
<b>getVoo()</b>	Retorna o número do voo
<b>getData()</b>	Retorna a data do voo
<b>clone()</b>	Efetua o clone de si próprio (objeto). Para isso, deve ser criado um novo objeto da mesma classe com os mesmos valores que o objeto original possui.

Ao construir o código, identifique as situações dentro dos métodos que podem violar as regras do negócio da classe e utilize exceções para sinalizar a violação, com a devida mensagem. Crie uma classe **Main.py** para servir de programa principal, testar os objetos e tratar as exceções adequadamente.

- 2) Considere a estrutura da classe **Gabarito** a seguir, cujo método **respostaQuestao()** recebe como parâmetro o número de uma questão e retorna a sua resposta correta, proveniente de um gabarito.

```
public class Gabarito {

    public char respostaQuestao(int numeroQuestao) {

        ...

    }

}
```

Codifique uma classe **Prova** em que cada objeto representa uma prova feita por um aluno. Esta prova possui 15 questões de múltipla escolha (letras A a E). As 10 primeiras questões valem 0,5 ponto e as 5 últimas questões valem 1 ponto. Esta classe deverá controlar as questões respondidas pelo aluno. Para isto, a classe deve implementar os métodos:

<b>construtor()</b>	Recebe como parâmetro o nome do candidato e um objeto da classe Gabarito contendo o gabarito da prova
<b>respostaAluno()</b>	recebe como parâmetro a resposta dada pelo aluno a uma questão; este método não recebe entre os parâmetros o número da questão, ele mesmo deve estabelecer um controle interno de modo que as questões sejam inseridas sequencialmente, ou seja, a primeira vez que o método é chamado, insere a primeira questão, a segunda, insere a segunda questão, e assim por diante.
<b>acertos()</b>	Retorna a quantidade de questões que o aluno acertou
<b>nota()</b>	Retorna a nota que o aluno tirou na prova
<b>comparaProva()</b>	Recebe como parâmetro um outro objeto da classe Prova e exibe na tela um resumo do desempenho de ambos os candidatos. O objeto prova que recebe a chamada do método <b>comparaProva()</b> será identificado como "host". As informações a serem exibidas são as seguintes:  <b>Alex (host):</b> nº questões certas = 14 nota: 9.5  <b>Fulano:</b> nº questões certas = 12 nota: 8.0  >>> Alex, você tirou uma nova melhor do que Fulano
<b>alteraGabarito()</b>	Recebe como argumento um novo gabarito que será utilizado para realizar a correção da prova