

A lista deve ser entregue até o dia **15/03/2020**, às 23h59, no Moodle, os arquivos devem ser compactados em um arquivo *.zip* ou *.tar*. O arquivo compactado deverá conter o projeto Eclipse ou Netbeans, ou apenas os fontes em *.java*, da lista. **Não serão aceitos projetos com os códigos-fonte no formato *.class*!**

## Lista 2 - Conceitos Básicos de POO

Para **nenhum** dos exercícios abaixo é necessário implementar um sistema em três camadas. Apenas implemente as classes descritas.

1. (1 ponto) Defina as classes abaixo contendo ao menos três atributos que os representam adequadamente. Observe o Exemplo 1 da classe Livro (não é necessário implementar nenhum método além dos getters e setters);
  - Pessoa
  - Comida
  - Animal
  - Cidade
  - Filme
2. (1 ponto) Para cada uma das classes descritas acima, instancie pelo menos três objetos distintos de cada uma delas e exiba no console o retorno do método **toString()**. Implemente o método **toString()** para cada uma das classes conforme o Exemplo 1

---

```
class Livro{  
    private int codigo;  
    private String nome;  
  
    public String toString(){  
        return "Nome: " + this.nome + "\n" +  
               "Codigo: " + this.codigo;  
    }  
}
```

---

Exemplo 1: Exemplo de método toString

3. (1 ponto) Crie uma classe que modele um carro, que utilize outras 5 classes que modelem as principais partes dele. Por exemplo, um carro possui rodas, tanque de combustível, acentos, vidros, etc. Esses objetos devem possuir pelo menos 3 atributos que os definem (dimensões, quantidades, etc).

4. (2 pontos) Implemente duas classes em Java para representar uma turma de estudantes, uma das classes deve ser o Aluno e a outra deve ser a Turma. Cada aluno tem um nome e cinco notas e deve possuir também um método para calcular a média do aluno. A classe Turma tem um número  $n$  de alunos definido no construtor e deve conter um método para adicionar alunos na turma. A classe Turma também deve possuir um método para listar os alunos aprovados (média acima de 7,0).
5. (2,5 pontos) Crie uma classe Carta contendo os atributos que desejar. Implemente outra classe chamada Baralho que contém um número  $n$  de cartas (podendo ser fixo). A classe Baralho deve conter o método **embaralhar()** responsável por distribuir as  $n$  cartas de forma aleatória.
6. (2,5 pontos) Registrador é o termo utilizado para denominar a memória dentro da CPU que armazena um número limitado de  $n$  bits. Basicamente, ele pode ser representado como um vetor de 0's e 1's de  $n$  posições. Uma CPU realiza várias operações lógicas internas com seus registradores, sendo as mais comuns AND e OR.

Implemente uma classe que represente um Registrador e uma classe que represente a CPU, a classe CPU deve realizar operações de OR e AND com dois registradores e armazenar o resultado em outro registrador. Utilize 8 como sendo o número  $n$  de bits de um registrador e também como sendo o número total de registradores. As Figuras 1 e 2 apresentam exemplos dessas operações.

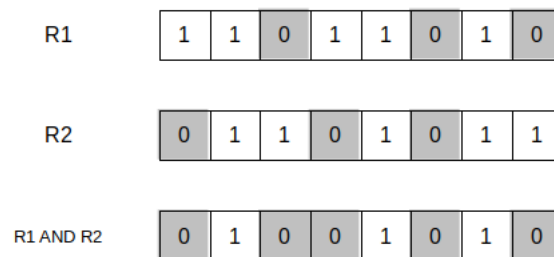


Figure 1: Operação AND entre dois registradores

A classe CPU deve ter os seguintes métodos:

- **void addRegistrador(Registrador registrador):** esse método adiciona o registrador ao array de registradores da cpu;
- **void getRegistradores():** esse método retorna os registradores da cpu;
- **Registrador operacaoOr(Registrador r1, Registrador r2);** esse método irá fazer  $r3 = r1 \text{ or } r2$
- **Registrador operacaoAnd(Registrador r1, Registrador r2);** esse método irá fazer  $r3 = r1 \text{ and } r2$

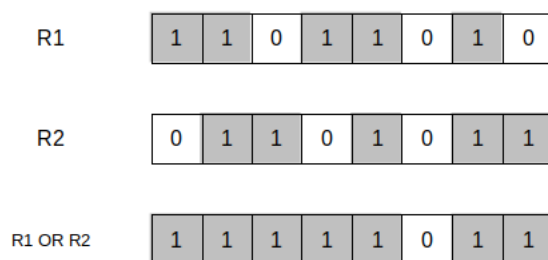


Figure 2: Operação OR entre dois registradores