

Questão 01:

Após instanciar objetos de todas as classes e adicioná-los ao baralho, ao imprimirmos as informações das cartas recebemos o seguinte output do console:

```
Lacaio(ID = 0)
Custo de Mana = 3
Ataque = 0
Vida Atual = 0
Vida Maxima = 0
```

```
Buff(ID = 0)
Custo de Mana = 2
```

```
Dano(ID = 0)
Custo de Mana = 4
dano = 2
```

```
DanoArea(ID = 0)
Custo de Mana = 10
dano = 11
```

```
magia(ID = 0)
Custo de Mana = 2
```

```
null
```

Questão 02:

A instanciação de um objeto do tipo carta não faz sentido neste exemplo. Visto que, carta é muito genérico em âmbito de relacionamento com o usuário. Isto é, o usuário (jogador) sempre terá contato com objetos mais específico, ou seja, algum tipo específico de carta.

Questão 03:

O método "usar(List<Carta> alvo)" não está sobrescrevendo qualquer outro método usar. Isso porque seu tipo de parâmetro é diferente dos demais apresentados, logo ele está apenas sobrecarregando a função usar.

A fim de comparação, segue abaixo a maneira como o método "usar" está implementado nas demais classes:

```
usar Carta alvo
```

Questão 04:

Os métodos de getters e setters da classe mãe são a única maneira das subclasses manipular os atributos quando estes têm visibilidade setada como "private". Logo, não é possível acessar o atributo nome diretamente na classe Lacaio.

Questão 05:

O método estático criado na classe "Carta" só poderia ser acessado pela subclasse se sua visibilidade estivesse setada para "public", caso contrário ele não seria visível pelas subclasses e receberíamos o seguinte erro:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The method Oi() from the type Carta is not visible
```

at Main.main(Main.java:31)

Questão 06:

A implementação da herança no código em questão traz uma série de benefícios:

1. Economia e organização de código, já que todos os tipos de carta possuem atributos em comum, haveria uma quantidade comum de código que se repetiria.
2. Facilidade para a implementação de possíveis mudanças, afinal, mudanças que venham a abranger todos os tipos de cartas poderão ser feitas com apenas uma modificação.