

08

Boleto bancário

Eduardo tem que lidar com a geração de boleto bancário para diversos bancos. Contudo, apesar dos boletos serem muito parecidos, cada banco possui um cabeçalho diferente.

Ele decidiu então escrever o seguinte código:

```
class Boleto {  
  
  geraLinhaDigitavel(): string {  
  
    /* lógica comum dos bancos */  
  }  
  
  geraCabecalho(): string {  
  
    throw new Error('Você precisa implementar a cabeçalho');  
  }  
}  
  
class BoletoBancoA extends Boleto {  
  
  geraCabecalho(): string {  
  
    /* lógica de geração do cabeçalho do banco A */  
  }  
}  
  
class BoletoBancoB extends Boleto {  
  
  geraCabecalho(): string {  
  
    /* lógica de geração do cabeçalho do banco B */  
  }  
}
```

Marque a afirmativa verdadeira a respeito do código de Eduardo.



Não faz sentido haver instâncias de `Boleto`, pois a classe não define a implementação de `geraCabecalho()`. Essa responsabilidade é das classes filhas, mas nada obriga o desenvolvedor a implementá-las em tempo de desenvolvimento e só será avisado caso tenha esquecido de implementá-lo em tempo de execução, no runtime da aplicação.



Correto!



As classes filhas são obrigadas em tempo de desenvolvimento a implementarem `geraCabecalho()`, caso contrário haverá um erro de compilação.



C

A classe `Boleto` precisa implementar a lógica de geração do cabeçalho para que seja utilizada pelas classes filhas.



Eduardo poderia ter transformado a classe `Boleto` uma classe abstrata, tornando `geraCabecalho()` abstrato. Além de não permitir que instâncias de `Boleto` sejam criadas em tempo de compilação (desenvolvimento), obrigará todas as filhas a implementarem `geraCabecalho()`. Enquanto não for implementado, as classes filhas não compilarão.

[PRÓXIMA ATIVIDADE](#)