

Será que o Miguel entendeu a aula?

Miguel estudou atentamente essa aula e fez as seguintes anotações abaixo. Você, estudioso que é, pode indicar quais afirmações Miguel escreveu equivocadamente através das alternativas abaixo?

1. Existe uma hierarquia grande de classes que representam exceções. Por exemplo, `ArithmeticException` é filha de `RuntimeException`, que herda de `Exception`, que por sua vez é filha da classe mais ancestral das exceções, `Throwable`. Conhecer bem essa hierarquia significa saber utilizar exceções em sua aplicação.
2. `Throwable` é a classe que precisa ser estendida para que seja possível jogar um objeto na pilha (através da palavra reservada `throw`).
3. A hierarquia iniciada com a classe `Throwable` é dividida em **exceções** e **erros**. Exceções são usadas em códigos de aplicação. Erros são usados exclusivamente pela máquina virtual. Classes que herdam de `ExceptionError` são usadas para comunicar erros na máquina virtual. Desenvolvedores de aplicação não devem criar erros que herdam de `Error`.
4. `StackOverflowError` é um erro da máquina virtual para informar que a pilha de execução não tem mais memória.
5. Exceções são separadas em duas grandes categorias: aquelas que são obrigatoriamente verificadas pelo compilador e as que não são verificadas. As primeiras são denominadas *checked* e são criadas através do pertencimento a uma hierarquia que não passe por `RuntimeException`. As segundas são as *unchecked*, e são criadas como descendentes de `RuntimeException`.

Selecione uma alternativa

A

Miguel se equivocou na afirmação 3.

B

Nenhuma. Todas as afirmações de Miguel estão corretas.

C

Miguel se equivocou nas afirmações 3 e 5.