Lucas Henrique – UC17201125

Exceções e controle de erros

Tem o intuito de melhorar tratamento de erros na aplicação, é comum que quem trate o erro é quem chamou o método e não a própria classe. Uma exceção representa uma situação que normalmente não ocorre e representa algo de estranho ou inesperado no sistema.

Um aspecto importante para os programadores é observar o rastro da pilha (*stacktrace*), apresenta o caminho que foi percorrido até ocorrer o erro, o sistema de exceções no Java funciona assim: quando é lançada uma exceção (*throw*), a JVM fica em estado de alerta, tentando tomar precaução ao tentar executar o trecho de código. As exceções são tratadas utilizando try/catch, onde vai ser tentando executar, caso ocorra um erro, ele será pego.

Existem dois tipos de exceptions: unchecked e checked. As unchecked são acusadas apenas em tempo de execução, o compilador não checa se o programador está tratando. As checked são aquelas que o compilador lhe obriga a tratar a exceção, assim o erro será checado em tempo de compilação, por meio da palavra reservada throws, a responsabilidade do tratamento ficará por conta de quem chamar o método. Declarar no throws as exceptions que são unchecked é desnecessário, mas não proibido, pois pode facilitar a leitura e documentação do código.

Utilizar o throw new é considerado lançar uma exceção unchecked, não obrigando o tratamento em tempo de compilação, utilizando o try/catch para pegar um possível erro. É possível criar nossas próprias classes exceptions para especificar ainda mais os tipos de erros, apenas herdando da classe escolhida.

Para finalizar, existe a funcionalidade finally, que sempre será executada no bloco try/catch, independentemente do resultado.