No java, uma exceção é um objeto herdado da classe:

-> java.lang.Exception – o compilador obriga a tratar ou propagar

-> java.lang.RuntimeException – o compilador não obriga a tratar ou propagar

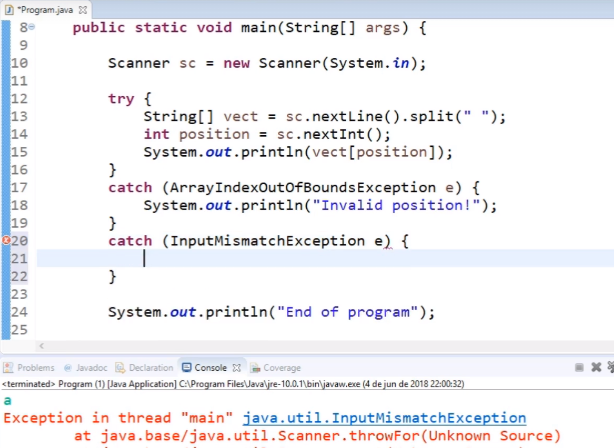
Try{

}

Catch (ExceptionType e){

}

No parâmetro do catch põe o tipo da exceção a tratar e um apelido pra ela, nesse caso: “e”.

Quando o Eclipse encontra um erro, expõe seu tipo no console, então basta criar um bloco catch com esse tipo de erro no parâmetro. No exemplo da imagem, no bloco “try” é possível ocorrer 2 erros: um por digitar número fora do alcance do vetor e outro por digitar letra ao invés de número.

Toda classe em java é uma subclasse de object, e este possui os métodos:

getClass: retorna o tipo do objeto

equals: compara se um objeto é igual a outro

hashCode: retorna um código hash do objeto

toString: converte o objeto para string

@Override: sobreposição, quando manda printar um objeto, printa o que está no override.

Atributo static: não instancia novo atributo a cada novo objeto criado.

Sobrecarga: recurso de uma classe para oferecer mais de uma operação de mesmo nome mas com parâmetros distintos.

Enumeradores: são um tipo especial que serve para especificar de forma literal, um conjunto de constantes relacionadas. Vantagens: melhor semântica, código mais legível e auxiliador pelo compilador.

Upcasting: casting/conversão da subclasse para superclasse (“subindo” da subclasse para superclasse). Esse tipo pode ser feito automaticamente

Uso comum: em polimorfismo.

Downcasting: casting da superclasse para subclasse. Precisa ser feito manualmente inserindo o tipo da subclasse entre parênteses e então, o nome do objeto da superclasse.

Uso comum: em métodos que recebem parâmetros genéricos.

Esse casting faz uma variável de um tipo de classe receber um objeto de outra classe.

Upcasting e Downcasting só funciona entre classes com herança entre elas. Upcasting é automático, mas o Downcasting, além de manual, o compilador não consegue saber se um downcasting é permitido até que tente fazê-lo, e isso pode resultar em erro. Para testar isso, podemos usar “instanceof”, exemplo: if( objeto1 instanceof classe)