



Nota: 10.0



Tempo: Terminou



Sair

Laboratório 8: Exceções

Introdução

A empresa privada [SpaceX](#) anunciou que irá enviar pessoas a Marte até o ano de 2028 (em 5 anos). Já a agência espacial [NASA](#) anunciou que irá enviar pessoas a Marte até 2030 (em 7 anos). Um passo importante em ambos os casos é encontrar o local mais apropriado em Marte para começar a colonização. Para isso, rovers como [Curiosity](#), [Perseverance](#) e [Ingenuity](#) são usados para encontrar tais locais (dentre outros objetivos). Tais rovers são semiautomáticos, no sentido que um caminho básico (lista de coordenadas) é enviado para eles a partir da Terra e eles seguem o caminho tomando decisões locais quando necessário. Entretanto, como os dados enviados da Terra seguem um meio de comunicação muito instável, medidas precisam ser tomadas para tratar imprevistos como coordenadas inválidas, caminhos inválidos, etc.



Figura 1: Auto-retrato do rover Curiosity em Marte

Objetivo

- Neste trabalho, você implementará uma hierarquia de [classes de exceções](#) para representar possíveis problemas em um caminho enviado para um rover.
- Além das classes, você irá declarar/implementar métodos que podem *disparar* exceções ([throws/throw](#)) bem com irá usar tais métodos *capturando* ([try](#)) e *tratando* ([catch](#)) as possíveis exceções.

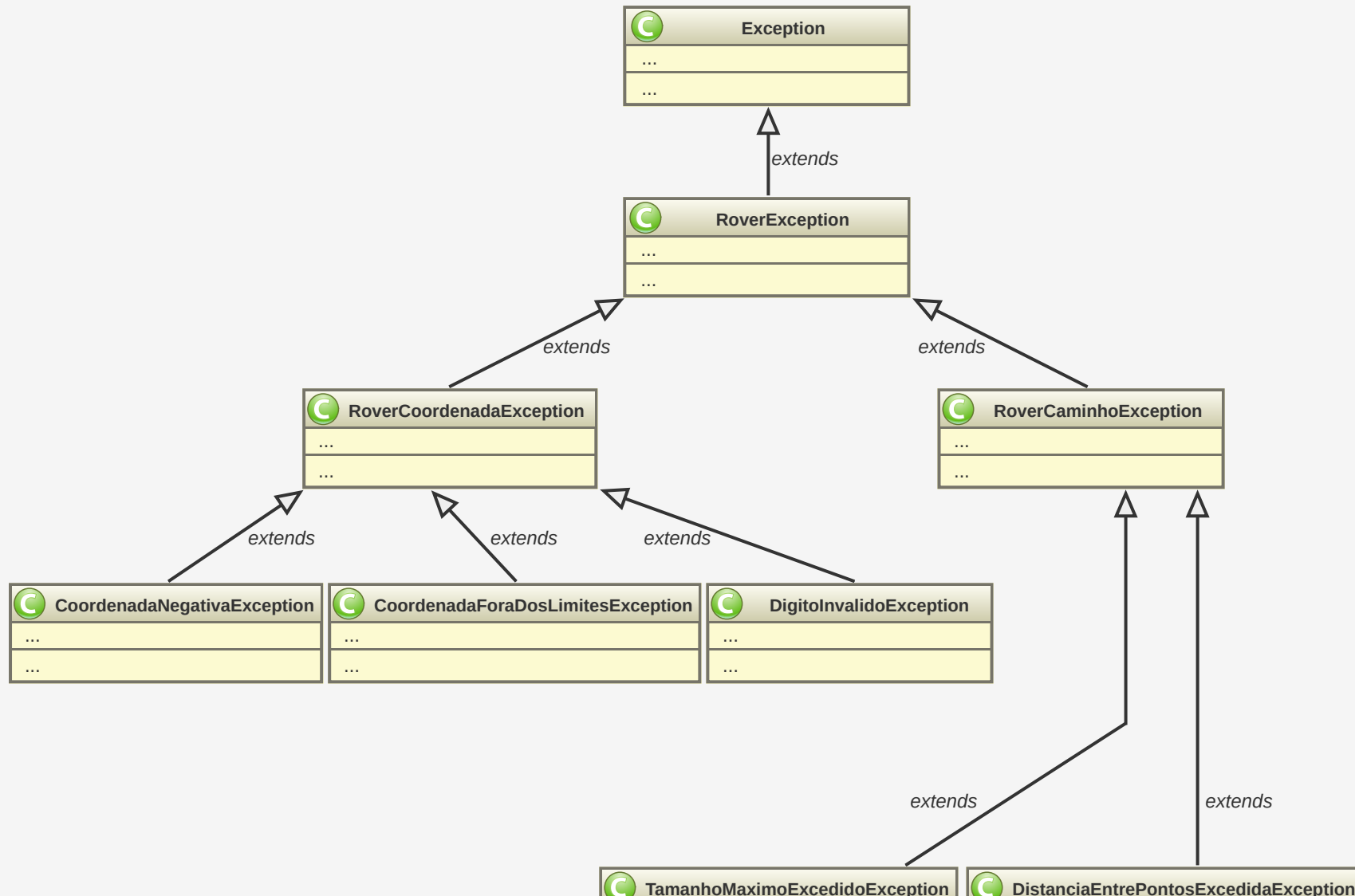
Informações

- Use os slides do arquivo "" como exemplos de código em Java para resolver este laboratório.
- Nome do pacote para todas as classes: `br.edu.ufam.icomp.lab_excecoes`

Questão 1: Classes de Exceções

Crie uma hierarquia de classes de exceções conforme figura abaixo. Estas classes serão usadas pelas classes das questões seguintes.

Atenção: Use o último exemplo dos slides para criar essas classes.



...	...
...	...

Cada classe de exceção possui uma "mensagem", que será impressa quando ela ocorrer. Seguem as mensagens das classes:

- `RoverException`: "Exceção geral do rover"
- `RoverCoordenadaException`: "Exceção geral de coordenada do rover"
- `RoverCaminhoException`: "Exceção geral de caminho do rover"
- `CoordenadaNegativaException`: "Coordenada com valor negativo"
- `CoordenadaForaDosLimitesException`: "Coordenada com valores fora dos limites"
- `DigitoInvalidoException`: "Digito da coordenada inválido"
- `TamanhoMaximoExcedidoException`: "Quantidade máxima de coordenadas excedida"
- `DistanciaEntrePontosExcedidaException`: "Distância máxima entre duas coordenadas vizinhas excedida"

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "RoverException.java"

Enviar "RoverCoordenadaException.java"

Enviar "RoverCaminhoException.java"

Enviar "CoordenadaNegativaException.java"




Enviar "CoordenadaForaDosLimitesException.java"


Enviar "DigitoInvalidoException.java"


Enviar "TamanhoMaximoExcedidoException.java"


Enviar "DistanciaEntrePontosExcedidaException.java"

Questão 2: Classe Coordenada


 Coordenada
 posX: <i>int</i>
 posY: <i>int</i>


 `digito: int`

 `Coordenada(int posX, int posY, int digito)`

 `getPosX(): int`

 `getPosY(): int`

 `distancia(Coordenada coordenada): double`

 `toString(): String`

Crie uma classe para representar uma **Coordenada**. O construtor da classe deverá verificar os dados passados e *disparar* as seguintes exceções caso seja necessário (em ordem de prioridade):

- **CoordenadaNegativaException**: esta exceção deverá ser disparada caso alguma das coordenadas seja menor que zero.
- **CoordenadaForaDosLimitesException**: disparada quando alguma das coordenadas não estiver entre os valores 0 e 30000.
- **DigitoInvalidoException**: disparada quando o resto da divisão da soma das coordenadas ($posX + posY$) por 10 for diferente do dígito (que deverá estar entre 0 e 9).

O método `distancia(Coordenada coordenada)` deverá retornar a distância entre a coordenada atual e a do argumento.

Já o método `toString` deverá imprimir as coordenadas de acordo com o exemplo abaixo:

| 32, 67

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Coordenada.java"

Solução correta!

Questão 3: Classe Caminho



Caminho