1)

- IllegalArgumentException: Esta exceção ocorre quando um método é chamado com um argumento inválido.

- ClassNotFoundException: Esta exceção ocorre quando a JVM tenta carregar uma classe, mas a classe especificada não pode ser encontrada.

- NumberFormatException : Esta exceção ocorre quando você tenta converter uma String em um tipo numérico, como int ou double, mas a String não contém um formato numérico válido.

2) Quando nenhum exceção é lançada dentro de um bloco try em Java, o controle de execução passa diretamente para qualquer bloco finally associado (se presente). Depois que o bloco finally é executado (ou se não houver bloco finally), a execução continua com o código após o bloco try-catch (se houver) ou simplesmente continua com o código após o bloco try se não houver bloco catch. Se não houver bloco finally, e nenhuma exceção for lançada, o controle passa diretamente para o código após o bloco try-catch.

3) A vantagem fundamental de utilizar um bloco catch em Java é a capacidade de capturar e tratar exceções que ocorrem dentro do bloco try. Isso permite que você escreva código para lidar com situações excepcionais de uma maneira controlada e previsível, em vez de permitir que as exceções interrompam abruptamente a execução do programa.

4) Se nenhum bloco catch é executado quando ocorre uma exceção dentro de um bloco try, a exceção é passada para o método chamador. Se não for tratada lá, ela é passada para o método chamador do método chamador e assim por diante, até que seja tratada ou até chegar ao método main(). Se não for tratada até o main(), geralmente resultará em uma mensagem de erro no console e a terminação do programa.

5) os blocos finally ajudam a garantir a execução de código importante que precisa ser executado, independentemente de ocorrer ou não uma exceção, melhorando a robustez e a segurança do programa.

6) a instrução throw serve para interromper a execução normal do programa e sinalizar que ocorreu um problema, permitindo que você forneça informações detalhadas sobre o erro para serem tratadas em algum lugar apropriado do código (como em um bloco catch ou em métodos chamadores).

7) Bloco try: Contém o código que pode lançar exceções.

Bloco catch: Captura e trata exceções que ocorrem dentro do bloco try.

Bloco finally: Contém o código que é sempre executado, independentemente de ocorrer ou não uma exceção no bloco try.

8) import java.util.Scanner;

```
public class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      int valor = 0;
      boolean entradaValida = false;

      while (!entradaValida) {
         try {
            System.out.print("Digite um valor inteiro: ");
            String entrada = scanner.nextLine();
            valor = Integer.parseInt(entrada);
            entradaValida = true; // Se a conversão para int for bem-sucedida, definimos entradaValida como true para sair do loop
         } catch (NumberFormatException e) {
            System.out.println("Erro: Por favor, digite um valor inteiro válido.");
         }
      }

      System.out.println("Você digitou o valor inteiro: " + valor);
   }
}
```