```java

public class GenericsB1A1<T, E, N, S, U , VIU>{
    /**
     * @Faculdade: Universidade Tecnológica Federal do Paraná
     * @Curso: Especialização em Tecnologia Java
     * @Nome: Erik Eduardo Valcezio RA: 02329611
     * @DATA: 13/07/2021
     */

    /* 1. Quais são as motivações para o uso dos genéricos em Java? */

    /**
     * Reaproveitamento de código (Reusabilidade); Coersões sendo implícitas
     * (Segurança de tipos); Os dois pontos estão ligados com a (Confiabilidade).
     */

    /*-2. Um tipo primitivo pode ser usado como argumento de tipo? */

    /**
     * Não, devido ao encapsulamento de tipos, sendo assim só aceitam classes
     * concretas, ou seja, objetos.
     */

    /*-3. Mostre como declarar uma classe chamada TesteDoisParametros que use dois parâmetros genéricos.*/

    class TesteDoisParametros<T, V> {
    }

    /*-4. Agora, altere TesteDoisParametros para que seu segundo parâmetro de tipo seja subclasse do
    primeiro parâmetro de tipo.*/

    class TesteDoisParametros2<T, V extends T> {
    }

    /*-5. No que diz respeito aos genéricos, o que é o símbolo "?" e o que ele faz?*/

    /**
     * Representa um tipo desconhecido, ou seja, um curinga equivale a qualquer
     * outro objeto válido em uma classe.
     */

    /*-6. O argumento curinga pode ser limitado?*/

    /**
```

```
45      * Sim, derivando o tipo de paramêtro, tanto para sua superclasse quanto
  para
46      * sua subclasse. *
47      */
48
49     /*-7. Um método genérico chamado MeuGenerico( ) tem um parâmetro de
  tipo. Além disso, MeuGenerico( )
50      * tem um parâmetro cujo tipo é o do parâmetro de tipo.
51      * Ele também retorna um objeto desse parâmetro de tipo. Mostre como
  declarar MeuGenerico( ).
52      */
53     <T> T meuGenerico(T parmTp) {
54         return parmTp;
55     }
56
57     /*
58      * 8. Dada a interface genérica a seguir
59      *
60      * interface IGenericoIF<T, V extends T> { // ...
61      *
62      * mostre a declaração de uma classe chamada MinhaClasse que implemente
63      * IGenericoIF.
64      */
65
66     class MinhaClasse<T, V extends T> implements IGenericoIF<T, V>{}
67
68     /*
69      * 9. Dada uma classe genérica chamada Contador<T>, mostre como criar um
  objeto
70      * de seu tipo bruto.
71      */
72
73     Contador contador = new Contador();
74
75     /*
76      * 10. Como a linha a seguir pode ser simplificada?
77      * MinhaClasse10<Double,String> obj = new MinhaClasse10Double,String>
  (1.1,"Hi");
78      */
79
80     MinhaClasse10<Double, String> obj = new MinhaClasse10<>(1.1, "Hi");
81
82 }
83
84 /*complementos para atividades*/
85 interface IGenericoIF<T, V extends T> {}
86 class Contador<T>{}
87 class MinhaClasse10<T, V>{MinhaClasse10(T t, V v){}}
```