1. Defina:
   1. Interfaces
   2. Métodos abstratos
   3. Métodos estáticos
   4. Polimorfismo
2. Explique a função de cada palavra reservada em Java:
   1. **super**
   2. **static**
   3. **implements**
   4. **extends**
   5. **throws**
3. Analise o código abaixo, marque a **alternativa correta** e **justifique sua resposta**:

**class** A{

**public** **void** print() **throws** Exception {

System.***out***.println("Hello, A!");

}

}

**class** Test{

**public** **static** **void** main(String[] args) {

A a = **new** A();

a.print();

}

}

* + O código executa com sucesso e imprime “Hello, A!”
  + O código não compila
  + O código compila, mas ao executar lança exceção

|  |  |
| --- | --- |
| **Resposta:** |  |
| **Justificativa:** |  |

1. Analise o código abaixo, marque a **alternativa correta** e **justifique sua resposta**:

**class** A{

**public** **static** **void** print() {

System.***out***.println("Hello, A!");

}

}

**class** Test{

**public** **static** **void** main(String[] args) {

A a = **new** A();

a.*print*();

}

}

* + O código executa com sucesso e imprime “Hello, A!”
  + O código não compila
  + O código compila, mas ao executar lança exceção

|  |  |
| --- | --- |
| **Resposta:** |  |
| **Justificativa:** |  |

1. Analise o código abaixo, marque a **alternativa correta** e **justifique sua resposta**:

**interface** A{

**public** **void** print();

}

**class** B **implements** A{

}

**class** Test{

**public** **static** **void** main(String[] args) {

B b = **new** B();

b.print();

}

}

* + O código compila
  + O código não compila

|  |  |
| --- | --- |
| **Resposta:** |  |
| **Justificativa:** |  |

1. Analise o código abaixo, marque a **alternativa correta** e **justifique sua resposta**:

**interface** IA{

**void** print();

}

**class** A **implements** IA{

@Override

**public** **void** print() {

System.***out***.println("Hello, A!");

}

}

**class** Test{

**public** **static** **void** main(String[] args) {

A a = **new** A();

a.print();

}

}

* + O código executa com sucesso e imprime “Hello, A!”
  + O código não compila
  + O código compila, mas ao executar lança exceção

|  |  |
| --- | --- |
| **Resposta:** |  |
| **Justificativa:** |  |

1. Baseado no SOLID, marque Verdadeiro ou Falso:
   1. ( ) Módulos de alto nível não devem depender da implementação de módulos de baixo nível, mas sim de suas abstrações.
   2. ( ) É uma boa prática criar interfaces especificas para cada módulo / camada de uma aplicação visando reduzir o acoplamento entre elas.
   3. ( ) Cada classe ou módulo de um sistema deve ser coeso em resolver apenas uma responsabilidade específica.
   4. ( ) Alto acoplamento é uma boa característica de projetos orientados a objetos