PACKAGES, EXCEPTION HANDLING

SLIPTEST-3

05-11-2024

Each question is for 10 marks.

Refer textbook chapter 10 and 11, classnotes and reference link in class diary.

|  |
| --- |
| Define package. Explain the behaviour of different access specifiers in packages.  Definition of package.  Draw the access specifiers table and give explanation/  Explain with example program.  Refer 10.1.1,10.5 |
| Write a program to demonstrate the working of checked and unchecked exceptions.  Definition of checked and unchecked exceptions.  Explain both with an example program. |
| Develop a Java program by defining, creating and accessing a package.  Suppose you have a package named com.example.myapp. How would you import a class named MyClass from this package into another Java file?  Definition of packages.  Steps in creating and accessing packages.  Steps for the given problem and write a sample program for it.  Refer 10.1.1,10.2,10.2.1,10.3 |
| Write a program to demonstrate the working of user defined exceptions.  Explain why do we need a user defined exception.  Write a user defined exception program. |
| What is the purpose of the java.lang package in Java? Which classes are automatically imported from this package?  Explain the use of java.lang package.  Explain 5 classes in it and their significance and methods in it.  Refer 10.7 |
| Develop a java program that demonstrates how certain exception types are not allowed to be thrown.  In Java, there are some exception types that are checked exceptions and require explicit handling. However, you might be interested in knowing that Java disallows the throwing of instances of java.lang.Error or its subclasses in regular scenarios because they represent serious errors that a typical application should not catch or handle. Let's show a simple example:  public class Main {  public static void main(String[] args) {  try {  throw new IllegalArgumentException("This is an example of a checked exception.");  } catch (IllegalArgumentException e) {  System.out.println("Caught IllegalArgumentException: " + e.getMessage());  }  // Unchecked exceptions (like ArithmeticException) can be thrown without needing to be declared or caught.  throw new ArithmeticException("Unchecked exception example.");  // However, throwing an Error or any of its subclasses is not typical and usually discouraged.  // throw new StackOverflowError("Error example.");  }  }  In this program:   1. An IllegalArgumentException is thrown and caught. 2. An ArithmeticException (an unchecked exception) is thrown without needing a try-catch block. 3. The commented line shows how an Error can be thrown, but it's generally not recommended to catch such errors in typical application code.   Java encourages you to handle exceptions gracefully rather than throwing Error types. Want to dive deeper into error handling? |
| Explain the role of the Object class in Java. Why is it considered the root class for all other classes?  Refer 10.8  The Object class is foundational to Java's class hierarchy, holding the distinction of being the root class. Here's why:   1. **Universal Superclass**: Every class in Java, directly or indirectly, inherits from the Object class. This means all classes implicitly gain the methods defined in Object, like toString(), equals(), hashCode(), and clone(). 2. **Method Consistency**: It provides a standard set of methods that every Java object can call, ensuring consistency and predictability. For example, the equals() method allows objects to be compared for equality, and hashCode() is used in hashing-based collections like HashMap. 3. **Polymorphism and Interfaces**: With Object as the root, it supports polymorphism, a key feature in Java. Methods that use Object as a parameter type can accept any object, making Java versatile and flexible. 4. **Garbage Collection**: The finalize() method, though deprecated, was originally part of Object to help with garbage collection cleanup.   By establishing a common superclass, Java ensures a unified and reliable structure, essential for creating robust, maintainable code.  Explain any few specific method in the Object class particularly useful |
| Develop a java program that illustrates the application of multiple catch statements.  Explain the significance of multiple catch clauses.  Write a program and explain it. |
| Give a brief description about the role of wrapper classes in java. Explain the concept of auto-boxing and auto-unboxing.  Explain the importance of Wrapper classes. In real world applications.  Explain the importance of Auto-boxing and auto-unboxing.  Write a program that includes all the above concepts.  Refer 10.11 and 10.12 and all subtopics |
| Demonstrate the class Throwable with the help of a Java program.  Refer 11.2 |
| Describe the significance of the Math class. What mathematical operations can you perform using its static methods?  Use and significance of Math class. Explain any 5 methods.  Refer 10.10 |
| What is an exception? Explain exception handling mechanism with an example.  Define exception.  Explain try,catch,finally,throw,throws  Write a sample program. |