**Template: Study Material**

|  |  |  |  |
| --- | --- | --- | --- |
| *Insert the details within the < quotes >*  <CO2>: <22412>: <Java Programming>: <Derived Syntactical Constructs in Java>: <LO5-C>: <Study Material> | | | |
| <Manish Pokharkar> | <13/02/2021 > | <Vijay Patil> |
| Key words  **Wrapper classes , enumerated types** | Learning Objective:  **student should understand the concept of wrapper classes and enumerated types** | Diagram/ Picture |
| Key Questions  Program to create the wrapper class object.  Ans:  public class WrapperDemo  {  public static void main(String[] args)  {  Integer myInt = 10;  Double myDouble = 11.65;  Character myChar = 'T';  Boolean myBool= true;  System.out.println(myInt);  System.out.println(myDouble);  System.out.println(myChar);  System.out.println(myBool);  }  }  Output:  10  11.65  T  True  Program for enumerated types in Java.  Ans:  class EnumExample1{  public enum Season { WINTER, SPRING, SUMMER, FALL }  public static void main(String[] args) {  for (Season s : Season.values())  System.out.println(s);  }} | **Concept Map**  Java Wrapper Class Hierarchy    **Concept Explanation:**  Vectors cannot handle primitive data types like int, float, long, char and double. Primitive data types may be converted into object types by using the wrapper classes.  The wrapper classes for numeric primitives have the capability to convert any valid number in string format into its corresponding primitive object. For example “10" can be converted into an Integer by using: Integer intVal = new Integer("10");  The wrapper object of a wrapper class can be created in one of two ways: by instantiating the wrapper class with the new operator or by invoking a static method on the wrapper class.  • Each of Java's primitive data types has a class dedicated to it. These are known as wrapper classes, because they "wrap" the primitive data type into an object of that class.  • The wrapper classes are part of the java.lang package, which is imported by default into all Java programs. |
| Simple data types and their corresponding wrapper class types are as follows:   |  |  | | --- | --- | | Simple type | Wrapper class | | boolean | Boolean | | char | Character | | double | Double | | float | Float | | int | Integer | | long | Long |   The wrapper classes have number of unique methods for handling primitive data types and objects, they are listed below:  Converting primitive numbers to object numbers using constructor methods   |  |  | | --- | --- | | Constructor calling | Conversion action | | Integer IntVal = new Integer(i); | Primitive integer to Integer objects. | | Float floatVal = new Float(f); | Primitive float to Float object. | | Double DoubleVal = new Double(d); | Primitive double to Double object. | | Long LongVal = new Long(l); | Primitive long to Long object. |   Note: i, f, d, and l are primitive data values denoting int, float, double and long data types. They may be constants or variables.  Converting object numbers to primitive numbers using typeValue() method   |  |  | | --- | --- | | Method calling | Conversion action | | int i = IntVal.intValue(); | Object to primitive integer. | | float f = FloatVal.floatValue(); | Object to primitive float. | | long l = LongVal.longValue(); | Object to primitive long. | | double d = DoubleVal.doubleValue(); | Object to primitive double. |   Converting numbers to Strings using to String() method   |  |  | | --- | --- | | Method calling | Conversion action | | str = Integer.toString(i); | Primitive integer to string. | | str = Float.toString(f); | Primitive float to string. | | str = Double.toString(d); | Primitive double to string. | | str = Long.toString(l) | Primitive long to string. |   Converting String objects to numeric objects using the static method ValueOf()   |  |  | | --- | --- | | Method calling | Conversion action | | DoubleVal = Double.Valueof(str); | Converts string to Double object. | | FloatVal = Float.ValueOf(str); | Converts string to Float object. | | IntVal = Integer.Valueof(str); | Converts string to Integer object. | | LongVal = Long.ValueOf(str); | Converts string to Long object. |   Converting numeric strings to primitive numbers using Parsing methods   |  |  | | --- | --- | | Method Calling | Conversion action | | int i = Integer.parseInt(str); | Converts string to primitive integer | | long i = Long.parseLong(str); | Converts string to primitive long |   Autoboxing and Unboxing  The *Autoboxing* and *Unboxing*  was released with the Java 5. During assignment, the automatic transformation of primitive  type (int, float, double etc.) into their object equivalents or wrapper type (Integer, Float, Double,etc) is known as *Autoboxing*. During assignment or calling of constructor, the automatic transformation of wrapper types into their primitive equivalent is known as Unboxing.  Example:  int i = 0;  i = new Integer(5); // auto-unboxing  Integer i2 = 5; // autoboxing | Key Definitions/ Formulas |
|  |
| Enumerated Types  J2SE 5.0 allows us to use the enumerated type in java using enum keyword. Enum type is a type which consists of fixed set of constant fields. This keyword can be used similar to the static final constants.  Example:  public class Days  {  public static final int Sunday=0;  public static final int Monday=1;  public static final int Tuesday=2;  public static final int Wednesday=3;  public static final int Thursday=4;  public static final int Friday=5;  public static final int Saturday=6;  }  The above code can be rewritten using enumerated are:  Public enum Day{Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}  Advantages:   * Compile time type safety. * We can use the enum keyword in switch statements. |  |
|  | Application of Concept/ Examples in real life |  |
| Key Take away from this LO:  Concept of array in Java | | | |