What is generic programming ? And how it is implemented in Java?

Generic Programming is style of programming language in which algorithms are written in such a way that its type is not specified. The type will be specified later in the time of execution. This increase the reusability in the code. We do not need to write the program with overloaded function (performing the same action on different data types).

How it is implemented in Java?

The following Generic class sum the sum number regardless of their type

package generic;

public class Generic {

public static <T extends Number > T sumNum(T x, T y )

{

T sum = x + y;

return sum;

}

public static void main(String[] args)

{

sumNum(10, 13);

sumNum(1.4, 15.6);

}

}

Differentiate between static binding and dynamic binding.

Static binding

It is done at the compile time. Static binding is use type of calls and fields and actual objects are not used in the binding. Method overloading is the example of Static binding. Private ,static and final method use the Static binding. It is also called early binding.

Dynamic binding

It is done at runtime. Dynamic binding uses actual objects to resolve the binding.

Method overriding is the example of Dynamic binding. A virtual method that can be redefine in the class is the example of Dynamic binding. It is also called early binding.

What are wrapper classes? When do you need to use wrapper classes in Java?

Wrapper Classes

Wrapper classes are those classed which are used to convert the primitive data type such as int, float, char, double into objects.

As Java is the Object-Oriented programming language, we do not let the Java to intestate the primate type variable as object because of performance issues. In OO model we do a lot with the primate type variables, which cannot be done in the primitive form. Interger.parseInt(str) is an example of the Java wrapper class.

Differentiate between mutable and immutable strings classes in Java.

Every String is immutable in Java. What is store in the String variable is just reference of the String. But in the String builder from we can say that the Strings are mutable.

* String builder is not thread safe but Strings is.
* + sign can be used to concatenate the string but in String builder form used the append method
* String builder is fatter but String have performance issues.
* String builder is good in multithreading but String is not
* String builder have some extra methods.

What is a major difference between checked and unchecked exceptions in Java?

The major difference is that unchecked exceptions are checked at runtime (while the program is running) while checked exceptions are checked at compile time. Checked exception can help us avoided the sudden termination of program, while unchecked exception can cause in sudden termination of program.

Explain the exception handling mechanism in Java.

Exception handing mechanism is Java mainly consists of five words such as try, catch, throw, throws, and finally. Exception are raised every time the user has done something wrong with program, to handle the unpredictive behavior of the user programmer use the exception handling. There are many types of exception that compiler knows but could not handle it self. In this process programmer use the try and catch block to handle the exception. Try block is something that tell the compiler that try this thing, if it does not work tell the user what they have done wrong using catch block. After that program can safely exit or give another change to user.

What are object serialization and deserialization?

Serialization

Is the process of converting the Object into the byte streams for the purpose of storing and transporting the object instance value in the file? We use the ObjectOutputStream class for this purpose which convert the object into two series of bytes.

Deserialization

Is the process of converting the byte streams received from the network or stored file data into Objects. We use the ObjectInputSteams class for this purpose which take data for byte streams and convert it into object.

What is the difference between Byte streams and Character stream classes in Java? Explain with appropriate examples.

Byte Streams

As the name suggest the byte steams are used to convert the data into byte (8-bit). The most used class for the byte streams is FileInputStreams which is used to convert the images and text into the byte streams.

Character Streams

As the name suggest the character streams are used to convert the data into 2-byte (16-bit) Unicode form. The most used class of character streams is File Reader and File Writer. The character streams are used to convert the character bytes into character and vice versa.

Differentiate between Array and Array List. When to use Array or Array List?

Array

* Array have fix length.
* Can be used in the primitive data type or object
* They have static nature
* Can be multidimensional

Array List

* Can be instantiated at the fix length but can be changed later
* Only be used to store the object
* The have dynamic nature
* Only signal dementia

Define a class named Payment that contains an instance variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named payment Details that outputs an English sentence to describe the amount of the payment.

Next, define a class named Cash Payment that is derived from Payment. This class should redefine the payment Details method to indicate that the payment is in cash. Include appropriate constructor(s).

Define a class named CreditCardPayment that is derived from Payment. This class should contain instance variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout.

Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package terminal;

/\*\*

\*

\* @author Arsla

\*/

public class Payment {

public double paymentAmount;

public Payment(double paymentAmount) {

this.paymentAmount = paymentAmount;

}

public double getPaymentAmount() {

return paymentAmount;

}

public void setPaymentAmount(double paymentAmount) {

this.paymentAmount = paymentAmount;

}

public void paymentDertail()

{

System.out.print("The amount of the Payment is "+ this.paymentAmount);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package terminal;

/\*\*

\*

\* @author Arsla

\*/

public class CashPayment extends Payment{

public CashPayment(double paymentAmount) {

super(paymentAmount);

}

@Override

public void paymentDertail()

{

System.out.print(super.paymentDertail()+" Rs.");

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package terminal;

/\*\*

\*

\* @author Arsla

\*/

public class CreditCardPayment extends Payment {

private String name;

private String expirationData;

private String creditCardNumber;

public CreditCardPayment(String name, String expirationData, String creditCardNumber, double paymentAmount) {

super(paymentAmount);

this.name = name;

this.expirationData = expirationData;

this.creditCardNumber = creditCardNumber;

}

@Override

public void paymentDertail()

{

System.out.println("Name\n"+this.name+" ExpirationDate"+this.expirationData+" Card Number"+this.creditCardNumber+super.paymentDertail());

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package terminal;

/\*\*

\*

\* @author Arsla

\*/

public class Test {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

CashPayment p1 = new CashPayment(1900d);

CashPayment p2 = new CashPayment(2000d);

CreditCardPayment c1 =new CreditCardPayment("Arslan", "30/12/2002", "13455", 4523d);

CreditCardPayment c2 =new CreditCardPayment("Hanan", "06/09/2004", "14354", 2356d);

}

}