Object Oriented Analysis and Design with Java

UE19CS353

Dr. L. Kamatchi Priya
Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

UE19CS353: Object Oriented Analysis and Design with Java

Generic Programming

Dr. L. Kamatchi Priya

Department of Computer Science and Engineering

Generic Programming

- Generics means parameterized types
- Enables to create classes, interfaces, and methods
- The data upon which they operate is specified as a parameter.
- Generic programming refers to writing code that will work for any data irrespective of their data type.
- Example: ArrayList is just one class, but the source code works for many different types. This is generic programming.
- An entity such as classes / interfaces / methods that works for any data types is known as generic entity
- By using generics, we can implement algorithms that work on different types of objects and at the same time, they are type safe too.

Generic Classes

Generic classes **encapsulate operations that are not specific to a particular data type**.
The most common use for generic classes is with collections like linked lists, hash tables, stacks, queues, trees, and so on.

To create an instance of generic

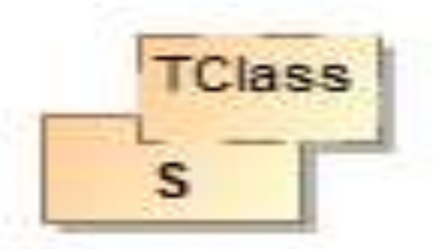
```
BaseType <Type> obj = new BaseType <Type>()
```

- <> to used to specify parameter types in generic class creation
- In Parameter type we can not use primitives like 'int','char' or 'double'.

UML representation of Generic Class, Interface and Method

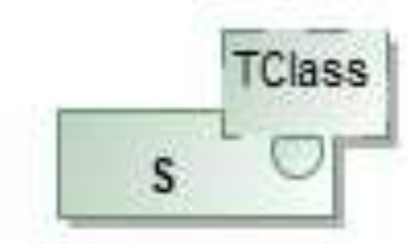
Generic Class

```
public class S<T>
{
}
```



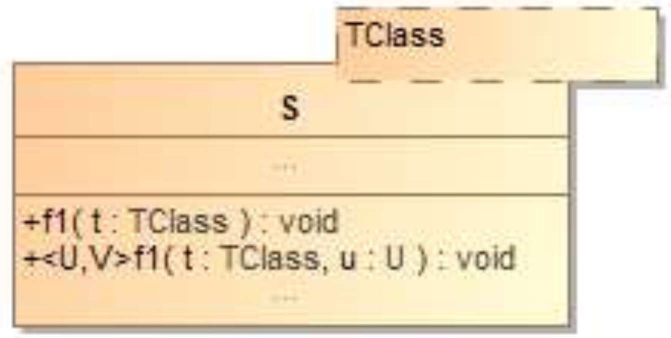
Generic Interface

```
interface S<T>
{
}
```



Generic Method

```
class b
{
 public T f1<T, U>(T t, U u)
       where U:T {return t;}
}
```



Generic Class implementation using Java

```
class Test<T>
{
      T obj;
      Test(T obj) { this.obj = obj; }
      public T getObject() { return this.obj; }
}

class Main
{
      public static void main (String[] args)
      {
            Test <Integer> iObj = new Test<Integer>(15);
            System.out.println(iObj.getObject());
            Test <String> sObj = new Test<String>(“OOADJ”);
            System.out.println(sObj.getObject());
      }
}
```

Generic Functions

Generic functions are called with different types of arguments based on the type of arguments passed to the generic method

```
class Test {
     static <T> void genericDisplay (T element){
          System.out.println ( element.getClass().
               getName()+" = " + element);
     }
     public static void main(String[] args){
          genericDisplay(11);
          genericDisplay(“OOADJ");
          genericDisplay(1.0);
     }
}
```

Concepts of Generic Programming

- Generics work only with Reference Types

- Generic types differ based on their type arguments

```
class Test<T> {
        T obj;
        Test(T obj) { this.obj = obj; } // constructor
        public T getObject() { return this.obj;
}
class Main {
        public static void main (String[] args) {
                Test <Integer> iObj = new Test<Integer>(15);
                System.out.println(iObj.getObject());

                Test <String> sObj = new Test<String>(“OOADJ");
                System.out.println(sObj.getObject());
                iObj = sObj; //This results in an error
        }
}
```

Advantages of Generics:

- Code Reuse
- Type Safety
- Individual Type Casting is not needed

Individual Type Casting:
When using generics, during compile time we can identify the data of collection is of same type or not; thus can avoid typecasting of individual elements of the collection

PES UNIVERSITY ONLINE

THANK YOU

Dr. L. Kamatchi Priya

Department of Computer Science and Engineering

priyal@pes.edu