```
-:- If we do type carring by specifying the errors class some then there is possibility of getting. ClassCartCoception
          Disject objects.gatBean("wiff"); // actually gives. Object ref pointing to WeekDayFinder class object.
                  //type casting
          LocalDate Seder (LocalDate) obj. // So type cacting with LocalDate does will throw ClassCostSecontion
*> To solve these problems and to make the code as type sale code: take the support of - 6 energies.

    • • Observes makes the code to flexible return types , parameteter types and etc...

    Moking the code as type safe code is, nothing but avoid type conting related. ClassCottException problem.

 upto spring 2.4
     public Object getBean(String beanid); //with out generic support
  eq: Object obj = ctc.outBean("wdf");
         WeekDayFinder finder=(WeekDayFinder) obj: //Since we are doing type carting here.
                                                                we can say the code is not type aris code
   From \Phi^{(i)} \Phi^{(i)} . (With Generics support, there is no possibility of getting type casting problem)
                                                                                                         ≈>javeJeng.Class is a pre-defined jave class
      public \quad \underline{<\mathbf{T}} \geq \quad \underline{\mathbf{T}} \quad getflear(String \ beautid, Class < \underline{\mathbf{T}}) \cdot dutz)
                                                                                                         The object of java langulass represents/holds.
             Type return.
                                                         cless name in the form of
                                                                                                          dass or interface or Enum or annotation or data
                                                         isvalanc. Care obi based which
             ded type
                                                                                                         type in a running java app-
                                                          the return of the method is
                                                          decided
                                                                                                         -> How numeric data type(int.long.float.double,...) variable
        hold numeric values. String variables hold text data
          (int)
Constitution/Planters states (int in Explications)
                                                                                                         Similarly the object of javadang Class can hold
         Web half the finite interests of the Cale ()
eg 2: Local Data date = cts.getDean ( Teleba", Local Data class);
                                                                                                        eksa ar interface or Enum ar data type ar annetation
                                                                               Here the code
                                                                               is type safe ande
         ogs: tocallime time-cta.getBean("ltime".Locallime.class):
                                                                                                        >The easiest way to create the object of
                                                                               herez of generics
                                                                                                        java lang. Class is using static "class" property
                                                        passing Local Time dass
                                                 bean.
                                                                                                         of every data given by java compiler dynamically
                                                           in the form of javalang-Class obj.
                                                           this makes the return type of
                                                                                                           Class of a Local Detection
                                                           etweetBeart, i in the Localline
                                                                                                             -> Gives - the object of java lang Class
                                                          since method returning, same Localli-
                                                                                                              having LocalBate and its metadata
                                                         date objithere is no insect of going for
                                                                                                               on the data of the object
ghingt of java long Class
Local Data store
                                                          any typecasting
                                                                                                                            LocalDate date
                                                                                                                            meja data
             janealung, Class

    "cines", "length" are built-in properties in

                                                                                                         эту јака арр
     >The object of this class holds given class/interface/Sourc/encoardon/data type metadata
                                                                                                             int all rese int (R10.20.30):
                                                                                                              Int also ta. length; //gives the size of the array

    There are insitiple ways to create this object expect using new operator.

     to Mises class is isseling private constructor , we can not create object for it using new
                                                                                                              Class c2: Week/DayFinder.class
      exervior autoide of the eless Definition.
                                                                                                                          obje<del>t jave</del> Jang Class
             (a) using gradition() method of jave languithjest class
                                                                                                                                     (Week Dav/Finder)
                   Date di new Desett
                                                                                                                                        class metadata)
                   Class of constant Classific
                                  jere<u>n karng Albana obj</u>
                                 ... dans dipo & lo mena dora
Colo alexe super dans
                                                      Date class implemented interfaces
                                                      Date class member variables
                                                       Date class method details:
                                                       Date days constrainting
                                                       and etc.
             b) by using ClassiforHame(-) method
                     \frac{\operatorname{disc obs}(\operatorname{configures}(\mathbb{T}_{\operatorname{permit}})) \operatorname{disc}(\mathbb{T}_{\operatorname{permit}})}{\operatorname{O}} 
                                  Loads the given "State" class and returns the objict
                                  jovallang. Good having "Date" class metadate as the
                                   data of the object
                                      autopaulian obj
             of upting Tolero" property ($454).

    This is compiler generated static property of every jews discussifype jews lang Claus

                      hald current class metadata information
                        Class classic total Data class.
```

Clarus, class S. Weschiffords volumes

elesco.

=> if we do type casting by specifying the wrong class name then there is possibility of getting ClassCastException

Object obj=ctx.getBean("wdf"); // actually gives Object ref pointing to WeekDayFinder class obj ref //type casting

LocalDate finder=(LocalDate)obj; // So type casting with LocalDate class will throw ClassCastException

- => To solve these problems and to make the code as type safe code take the support of Generics
- => Generics makes the code to flexible return types, parameteter types and etc...
- => Making the code as type safe code is nothing but avoid type casting related ClassCastException problem upto spring 2.4

public Object getBean(String beanid);

//with out generic support

eg:: Object obj = ctx.getBean("wdf");

WeekDayFinder finder=(WeekDayFinder) obj; //Since we are doing type casting here

we can say the code is not type safe code

From spring 2.5 (With Generics support, there is no possibility of getting type casting problem)

======

public <T> I getBean(String beanid, Class<⇒ clazz)

Type return

decl type

class name in the form of java.lang.Class obj based which the return of the method is decided

eg1:: WeekDayFinder finder=ctx.getBean("wdf",WeekDayFinder.class);

(or)

Class<WeekDayFinder> clazz=WeekDayFinder.class;

WeekDayFinder finder-ctx.getBean("wdf",clazz);

eg2:: LocalDate date=ctx.getBean("Idate", LocalDate.class); eg3:: LocalTime time=ctx.getBean("Itime", LocalTime.class);

java.lang.Class

\_\_\_\_\_

## bean

id

Here the code is type safe code becoz of generics passing LocalTime class in the form of java.lang.Class obj this makes the return type of ctx.getBean(-,-) as the LocalTime since method returning same LocalTime class obj there is no need of going for any typecasting

- =>The object of this class holds given class/interface/Enum/annotation/data type metadata i.e basic information + additional information
- => There are multiple ways to create this object expect using new operator
- => if java class is having private constructor, we can not create object for it using new operator outside of the class Definition

(a) using getClass() method of java.lang.Object class Date d=new Date(): Class clazz=d.getClass(); java.lang.Class obj =>java.lang.Class is a pre-defined java class =>The object of java.lang.class represents/holds class or interface or Enum or annotation or data type in a running java app => How numeric data type(int, long, float, double,...) variable hold numeric values, String variables hold text data Similarly the object of java.lang.Class can hold class or interface or Enum or data type or annotation =>The easiest way to create the object of java.lang.Class is using static "class" property of every class given by java compiler dynamically Class c1 = LocalDate.class; =>Gives the object of java.lang.Class having LocalDate and its metadata as the data of the object object of java.lang.Class LocalDate class meta data => "class", "length" are built-in properties in any java app int a[] = new int[]{10,20,30}; int size=a.length; //gives the size of the array Class c2=WeekDayFinder.class: c2 obj of java lang.Class WeekDayFinder class metadata) clazz date class & its meta data Date class super class Date class implemented interfaces Date class member variables Date class method details Date class constructors and etc. b) by using Class.forName(-) method Class clazz1=Class.forName("java.util.Date"); Loads the given "Date" class and returns the obj of java.lang.Class having "Date" class metadata as the data of the object java.lang.Class obj Date class metadata c) using "class" property (Best)

=>This is compiler generated static property of every java class of type java.lang.Class

```
hold current class metadata information
Class clazz2=LocalDate.class;
Class clazz3-WeekFinder.class;
clazz3
w.r.t spring programming
obj of java.lang.Class
LocalDate class metadata
AnnotationConfigApplicationContext ctx=new
AnnotationConfigApplicationContext(AppConfig.class);
Here we are not passing AppConfig class name, we are actually pssing the object of java.lang.Class having
AppCpnfig class meta data
Sample code giving method of certain using the methods invoked on the object of java.lang.Class
System.out.println(":
Class clazz1=WeekDayFinder.class;
System.out.println("class name::"+clazz1.getName());
=");
System.out.println("Suepr class name::"+clazz1.getSuperclass());
System.out.println("Implemented interfaces ::"+Arrays.toString(clazz1.getInterfaces()));
System.out.println("methods info ::"+Arrays.toString(clazz1.getDeclaredMethods()));
OUTput
class name::com.nt.sbeans.WeekDayFinder
Suepr class name::class java.lang.Object
Implemented interfaces :: [interface java.io.Serializable]
methods info::[public java.lang.String com.nt.sbeans.WeekDayFinder.showMessage(java.lang.String), public void
com.nt.sbeans.WeekDayFinder.setTime(java.time.LocalTime), public void com.nt.sbeans.WeekDayFinder.assign
Time(java.time.LocalTime), public void com.nt.sbeans.WeekDayFinder.setDate(java.time.LocalDate), public void
com.nt.sbeans.WeekDayFinder.putDate(java.time.LocalDate)]
=> Built-in threads in java app are :: main, gc (garbage collector)
=> Built-in streams in java app are :: System.in, System.out, System.err
=> Built-in properties in java app are :: class, length
=> Built-in reference variables/objs in java app are :: this, super
In java
======
=>"class" is a keyword
=>"class" is built-in property
=>"java.lang.Class" is pre-defined class
```

=>"class" is oops terminology