|  |
| --- |
| **public** **class** MyAction{  **private** String ad;  **private** String soyad;  **public** String getSoyad() {  **return** soyad;  }  **public** **void** setSoyad(String soyad) {  System.*out*.println(soyad);  **this**.soyad = soyad;  }  **public** **void** setAd(String ad) {  System.*out*.println("merhaba" +" " +ad);  **this**.ad=ad;  }  **public** String getAd() {  **return** ad;  }  } |

|  |
| --- |
| **import** java.lang.reflect.InvocationTargetException;  **import** java.lang.reflect.Method;  **public** **class** MetodOkuma {  /\*\*  \* **@param** args  \*/  **public** **static** **void** main(String[] args) {  **try** {  Class cls = Class.*forName*("MyAction");  Method[] methods = cls.getMethods();  **for** (**int** i = 0; i < methods.length; i++) {  System.*out*.println(methods[i].getName());  }  } **catch** (ClassNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

|  |
| --- |
| **import** java.lang.reflect.InvocationTargetException;  **import** java.lang.reflect.Method;  **public** **class** MetodCalistirma {  /\*\*  \* **@param** args  \*/  **public** **static** **void** main(String[] args) {  Class cls;  **try** {  cls = Class.*forName*("MyAction");  MyAction act = (MyAction) (cls.newInstance());  Method[] methods = cls.getMethods();  **for** (**int** i = 0; i < methods.length; i++) {  Method meth = methods[i];  **if** (meth.getName().startsWith("set"))  meth.invoke(act, "deneme");  }  } **catch** (ClassNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InstantiationException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalAccessException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalArgumentException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InvocationTargetException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

|  |
| --- |
| **public** **class** DortIslem {  **public** **int** add(**int** a, **int** b) {  **return** a + b;  }  **public** **int** sub(**int** a, **int** b) {  **return** a - b;  }  **public** **int** mul(**int** a, **int** b) {  **return** a \* b;  }  **public** **int** div(**int** a, **int** b) {  **return** a / b;  }  **public** **void** setFonk(String ad) {  System.*out*.println(ad);  }  } |

|  |
| --- |
| **import** java.lang.reflect.Method;  **public** **class** RunMthdRef1 {    **public** **static** **void** main(String[] args) {  **try** {  Integer[] input={**new** Integer(2),**new** Integer(6)};  Class cl=Class.*forName*("DortIslem");  Class[] par=**new** Class[2];  par[0]=Integer.*TYPE*;  par[1]=Integer.*TYPE*;    Class[] argTypes = **new** Class[] { String.**class** };  Method mthd=cl.getMethod("sub",par);  Method mthd1=cl.getMethod("setFonk",argTypes);    Integer output=(Integer)mthd.invoke(**new** DortIslem(),input);  System.*out*.println(output.intValue());  //mthd1.invoke(new RunMthdRef1(),"deneme");  } **catch** (Exception e) {  e.printStackTrace();  }  }  } |

|  |
| --- |
| public void validateEmail(FacesContext context, UIComponent toValidate,  Object value) throws ValidatorException {  String email = (String) value;  if (email.indexOf('@') < 0) {  FacesMessage message = new FacesMessage("Hatali mail");  throw new ValidatorException(message);  }  } |

|  |
| --- |
| **import** javax.faces.component.UIComponent;  **import** javax.faces.context.FacesContext;  **import** javax.faces.validator.ValidatorException;  **public** **class** MyAction{  **private** String ad;  **private** String soyad;  **public** String getSoyad() {  **return** soyad;  }  **public** **void** setSoyad(String soyad) {  System.*out*.println(soyad);  **this**.soyad = soyad;  }  **public** **void** setAd(String ad) {  System.*out*.println("merhaba" +" " +ad);  **this**.ad=ad;  }  **public** String getAd() {  **return** ad;  }    **public** **void** validateEmail(FacesContext context, UIComponent toValidate,  Object value) **throws** ValidatorException {  }    } |

|  |
| --- |
| **import** java.lang.reflect.Method;  **public** **class** MetodCalistirmaParam {  /\*\*  \* **@param** args  \*/  **public** **static** **void** main(String[] args) {  Class cls;  **try** {  cls = Class.*forName*("MyAction");  MyAction act = (MyAction) (cls.newInstance());  Method[] methods = cls.getMethods();  **for** (**int** i = 0; i < methods.length; i++) {  Method meth = methods[i];  Class[] params = meth.getParameterTypes();  Class[] except = meth.getExceptionTypes();  **if** (params.length == 3) {  **if** (params[0].getName().equals(  "javax.faces.context.FacesContext")  & params[1].getName().equals(  "javax.faces.component.UIComponent")  & params[2].getName().equals("java.lang.Object")  & except[0].getName().equals("javax.faces.validator.ValidatorException"))  System.*out*.println(meth.getName());  }  }  } **catch** (ClassNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InstantiationException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalAccessException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalArgumentException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |