**Lambdas FUNCTION CHAINING**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.util.ArrayList; **import** java.util.Random; **import** java.util.function.Function;  **public class** Main {   **public static void** main(String[] args) {  Employee snow = **new** Employee(**"Jon snow"**, 10);  Employee tygerrian = **new** Employee(**"Tygerrian Denerriys"**, 5);  Employee ned = **new** Employee(**"Ned Stark"**, 25);  Employee lenisters = **new** Employee(**"Kingslayer Drogo"**, 35);   ArrayList<Employee> employees = **new** ArrayList<>();  employees.add(snow);  employees.add(tygerrian);  employees.add(ned);  employees.add(lenisters);   Function<String, String> firstname = employee -> employee.substring(0, employee.indexOf(**' '**));  Function<Employee, String> upparcase = employee -> employee.getName().toUpperCase();  *// will throw an error because,  // firstname is a 'Function<String, String>'  // that cant pullup the name from Employee collection // Function concats = firstname.andThen(upparcase);* Function concatString = upparcase.andThen(firstname);  System.***out***.println(concatString.apply(employees.get(1)));  }   **public static** String getAname(Function<Employee, String> afunction, Employee emps){  **return** afunction.apply(emps);  } } | **TYGERRIAN** |

**BI FUNCTIONS -takes two parameters under lambdas**

CANT BE CHAINED

|  |  |
| --- | --- |
| **package** com.company;  **import** java.util.ArrayList; **import** java.util.Random; **import** java.util.function.BiFunction; **import** java.util.function.Function;  **public class** Main {   **public static void** main(String[] args) {  Employee snow = **new** Employee(**"Jon snow"**, 10);  Employee tygerrian = **new** Employee(**"Tygerrian Denerriys"**, 5);  Employee ned = **new** Employee(**"Ned Stark"**, 25);  Employee lenisters = **new** Employee(**"Kingslayer Drogo"**, 35);   ArrayList<Employee> employees = **new** ArrayList<>();  employees.add(snow);  employees.add(tygerrian);  employees.add(ned);  employees.add(lenisters);   Function<String, String> firstname = employee -> employee.substring(0, employee.indexOf(**' '**));  Function<Employee, String> upparcase = employee -> employee.getName().toUpperCase();  Function concatString = upparcase.andThen(firstname);  System.***out***.println(concatString.apply(employees.get(1)));   BiFunction<String, Employee, String> grabit = (String name, Employee emp) -> {  **return** name.concat(**" "** + emp.getExperience());  };   *//a bifunction cant be CHAINED, so in this case  //case 1 :* String uppered = upparcase.apply(employees.get(1));  String bifunctionCase = grabit.apply(uppered, employees.get(1));  System.***out***.println(**"From uppered = "** + bifunctionCase);   *//case 2 :* System.***out***.println(**" "**);  String uppered1 = employees.get(2).toString();  System.***out***.println(**"uppered1 = "** + uppered1);  String bifunctionCase1 = grabit.apply(uppered1, employees.get(2));  System.***out***.println(**"From uppered1 = "** + bifunctionCase1);  }   **public static** String getAname(Function<Employee, String> afunction, Employee emps){  **return** afunction.apply(emps);  } } | TYGERRIAN  From uppered = TYGERRIAN DENERRIYS 5    uppered1 = com.company.Employee@2812cbfa  From uppered1 = com.company.Employee@2812cbfa 25 |

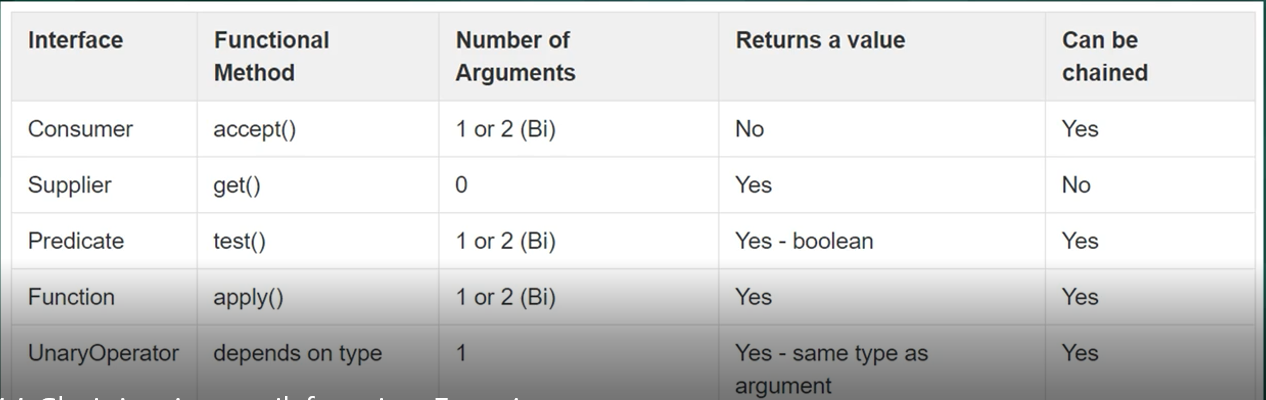
**Intuneryoperator**

|  |  |
| --- | --- |
| package com.company;  import java.util.function.Function; import java.util.function.IntUnaryOperator;  public class Main {   public static void main(String[] args) {  IntUnaryOperator intits = i -> i + 5;  System.*out*.println(intits.applyAsInt(10));  } } | 15 |

**Consumer<T>**

|  |  |
| --- | --- |
| package com.company;  import java.util.function.Consumer; import java.util.function.Function;  public class Main {   public static void main(String[] args) {  Consumer<String> c1 = c -> c.toUpperCase();  Consumer<String> c2 = c -> System.*out*.println(c);  c1.andThen(c2).accept("hello bitcH");  } } | hello bitcH |

**Types of interfaces**

****