**Lambda – FUNCTIONS**

**WITHOUT LAMBDA EXPRESSION**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.util.ArrayList; **import** java.util.List; **import** java.util.function.Predicate;  **public class** Main {   **public static void** main(String[] args) {  Employee snow = **new** Employee(**"Jon snow"**, 10);  Employee tygerrian = **new** Employee(**"Tygerrian Logi"**, 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);   employees.forEach(employee -> {  String lastname =  employee.getName().substring(employee.getName().indexOf(**' '**) + 1);  System.***out***.println(lastname);  });  } } | **snow**  **Logi**  **Stark**  **Drogo**  **package** com.company;  **class** Employee{  String **name**;  **int experience**;   *//constructor* **public** Employee(String name, **int** experience) {  **this**.**name** = name;  **this**.**experience** = experience;  }  *//getter and setter* **public** String getName() {  **return name**;  }  **public void** setName(String name) {  **this**.**name** = name;  }  **public int** getExperience() {  **return experience**;  }  **public void** setExperience(**int** experience) {  **this**.**experience** = experience;  } } |

**FUNCTIONS**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.util.ArrayList; **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 Logi"**, 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<T(type), R(return)>* Function<Employee, String> getthelastname = (Employee employee) ->{  **return** employee.getName().substring(employee.getName().indexOf(**' '**) + 1);  };   **for** (**int** i = 0; i < employees.size(); i++){  String lastnames = getthelastname.apply(employees.get(i));  System.***out***.println(lastnames);  }  } } | **snow**  **Logi**  **Stark**  **Drogo**  **package** com.company;  **class** Employee{  String **name**;  **int experience**;   *//constructor* **public** Employee(String name, **int** experience) {  **this**.**name** = name;  **this**.**experience** = experience;  }  *//getter and setter* **public** String getName() {  **return name**;  }  **public void** setName(String name) {  **this**.**name** = name;  }  **public int** getExperience() {  **return experience**;  }  **public void** setExperience(**int** experience) {  **this**.**experience** = experience;  } } |

**TWO DIFFERENT TASKS USING ONE METHODS: FUNCTION INTERFACE**

|  |  |
| --- | --- |
| **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<T(type), R(return)>* Function<Employee, String> getthelastname = (Employee employee) ->{  **return** employee.getName().substring(employee.getName().indexOf(**' '**) + 1);  };   **for** (**int** i = 0; i < employees.size(); i++){  String lastnames = getthelastname.apply(employees.get(i));  System.***out***.println(lastnames);  }   System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);   Function<Employee, String> getfirstname = (Employee employee) ->{  **return** employee.getName().substring(0, employee.getName().indexOf(**' '**));  };   **for** (**int** i = 0; i < employees.size(); i++){  String firstnames = getfirstname.apply(employees.get(i));  System.***out***.println(firstnames);  }   *//while USING BOTH case at a time;* System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);   Random r = **new** Random();  **for** (Employee emps : employees){  **if** (r.nextBoolean()){  *//function name, collection* System.***out***.println(*getAname*(getfirstname, emps));  } **else** {  System.***out***.println(*getAname*(getthelastname, emps));  }  }  }   **public static** String getAname(Function<Employee, String> afunction, Employee emps){  **return** afunction.apply(emps);  } } | **snow**  **Denerriys**  **Stark**  **Drogo**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **Jon**  **Tygerrian**  **Ned**  **Kingslayer**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **Jon**  **Denerriys**  **Stark**  **Kingslayer** |