1

**package** one;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.function.Consumer;

**import** java.util.function.Predicate;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** ChainConsumers {

**public** **static** **void** main(String... args) {

List<String> strings =

Arrays.*asList*("one", "two", "three", "four", "five");

List<String> result = **new** ArrayList<>();

Consumer<String> c1 = System.***out***::println;

Consumer<String> c2 = result::add;

System.***out***.println("c1");

strings.forEach(c1.andThen(c2));

System.***out***.println("size of result = " + result.size());

}

}

package one;

import java.util.Arrays;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

/\*\*

\*

\* @author JosÃ©

\*/

public class ComparatorLambda {

public static void main(String... args) {

// Comparator<String> comp = new Comparator<String>() {

//

// @Override

// public int compare(String s1, String s2) {

//

// return Integer.compare(s1.length(), s2.length());

// }

// };

Comparator<String> compLambda = (String s1, String s2) ->

Integer.compare(s1.length(), s2.length());

List<String> list = Arrays.asList("\*\*\*", "\*\*", "\*\*\*\*", "\*");

Collections.sort(list, compLambda);

for(String s : list) {

System.out.println(s);

}

}

}package one;

import java.io.File;

import java.io.FileFilter;

/\*\*

\*

\* @author JosÃ©

\*/

public class FirstLambda {

public static void main(String[] args) {

// FileFilter filter = new FileFilter() {

//

// @Override

// public boolean accept(File pathname) {

//

// return pathname.getName().endsWith(".java");

// }

// };

FileFilter filterLambda = (File pathname) ->

pathname.getName().endsWith(".java");

File dir = new File("d:/java");

File[] files = dir.listFiles(filterLambda);

for (File f : files) {

System.out.println(f);

}

}

}**package** one;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** RunnableLambda {

**public** **static** **void** main(String... args) **throws** InterruptedException {

// Runnable runnable = new Runnable() {

//

// @Override

// public void run() {

//

// for (int i = 0 ; i < 3 ; i++) {

// System.out.println(

// "Hello world from thread [" +

// Thread.currentThread().getName() +

// "]");

// }

// }

// };

Runnable runnableLambda = () -> {

**for** (**int** i = 0 ; i < 3 ; i++) {

System.***out***.println(

"Hello world from thread [" +

Thread.*currentThread*().getName() +

"]");

}

} ;

Thread t = **new** Thread(runnableLambda);

t.start();

t.join();

}

}

2

**package** stream;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.util.ArrayList;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Optional;

**import** java.util.Set;

**import** java.util.TreeSet;

**import** java.util.stream.Collectors;

**import** java.util.stream.Stream;

**import** stream.model.Person;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** CollectorsExample {

**public** **static** **void** main(String... args) {

List<Person> persons = **new** ArrayList<>();

**try** (

BufferedReader reader =

**new** BufferedReader(

**new** InputStreamReader(

CollectorsExample.**class**.getResourceAsStream("people.txt")));

Stream<String> stream = reader.lines();

) {

stream.map(line -> {

String[] s = line.split(" ");

Person p = **new** Person(s[0].trim(), Integer.*parseInt*(s[1]));

persons.add(p);

**return** p;

})

.forEach(System.***out***::println);

} **catch** (IOException ioe) {

System.***out***.println(ioe);

}

Optional<Person> opt =

persons.stream().filter(p -> p.getAge() >= 20)

.min(Comparator.*comparing*(Person::getAge));

System.***out***.println(opt);

Optional<Person> opt2 =

persons.stream().max(Comparator.*comparing*(Person::getAge));

System.***out***.println(opt2);

Map<Integer, String> map =

persons.stream()

.collect(

Collectors.*groupingBy*(

Person::getAge,

Collectors.*mapping*(

Person::getName,

Collectors.*joining*(", ")

)

)

);

System.***out***.println(map);

}

}

**package** stream;

**import** java.util.function.Predicate;

**import** java.util.stream.Stream;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** FirstPredicates {

**public** **static** **void** main(String[] args) {

Stream<String> stream = Stream.*of*("one", "two", "three", "four", "five");

Predicate<String> p1 = s -> s.length() > 3 ;

Predicate<String> p2 = Predicate.*isEqual*("two");

Predicate<String> p3 = Predicate.*isEqual*("three");

stream

.filter(p2.or(p3))

.forEach(s -> System.***out***.println(s));

}

}

**package** stream;

**import** java.util.Arrays;

**import** java.util.Collection;

**import** java.util.List;

**import** java.util.function.Function;

**import** java.util.stream.Stream;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** FlatMapExample {

**public** **static** **void** main(String... args) {

List<Integer> list1 = Arrays.*asList*(1, 2, 3, 4, 5, 6, 7);

List<Integer> list2 = Arrays.*asList*(2, 4, 6);

List<Integer> list3 = Arrays.*asList*(3, 5, 7);

List<List<Integer>> list = Arrays.*asList*(list1, list2, list3);

System.***out***.println(list);

Function<List<?>, Integer> size = List::size;

Function<List<Integer>, Stream<Integer>> flatmapper =

l -> l.stream();

list.stream()

.flatMap(flatmapper)

.forEach(System.***out***::println);

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package stream;

import java.util.ArrayList;

import java.util.List;

import java.util.function.Predicate;

import java.util.stream.Stream;

/\*\*

\*

\* @author JosÃ©

\*/

public class IntermediaryAndFinal {

public static void main(String[] args) {

Stream<String> stream = Stream.of("one", "two", "three", "four", "five");

Predicate<String> p1 = Predicate.isEqual("two");

Predicate<String> p2 = Predicate.isEqual("three");

List<String> list = new ArrayList<>();

stream

.peek(System.out::println)

.filter(p1.or(p2))

.forEach(list::add);

System.out.println("Done!");

System.out.println("size = " + list.size());

}

}package stream;

import java.util.Arrays;

import java.util.List;

import java.util.Optional;

/\*\*

\*

\* @author JosÃ©

\*/

public class ReductionExample {

public static void main(String... args) {

List<Integer> list = Arrays.asList();

Optional<Integer> red =

list.stream()

.reduce(Integer::max);

System.out.println("red = " + red);

}

}

People.txt

Sarah 15

Philip 21

Beth 23

Simon 24

Nina 23

Allan 28

Leonard 18

Barbara 26

Penelope 18

Albert 31

Lucy 42

Charles 26

Ella 24

Louis 32

Liza 17

Franck 18

Amy 26

Nathan 33

**package** stream.model;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** Person {

**private** String name;

**private** **int** age;

**public** Person(){}

**public** Person(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** String getName() {

**return** **this**.name;

}

**public** **int** getAge() {

**return** **this**.age;

}

**public** String toString() {

**return** "Person [" + **this**.name + ", " + **this**.age + "]";

}

}

3 Date and Time

//package org.paumard;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.time.LocalDate;

import java.time.Month;

import java.time.Period;

import java.time.format.DateTimeFormatter;

import java.time.temporal.ChronoUnit;

import java.time.temporal.TemporalUnit;

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Stream;

/\*\*

\*

\* @author JosÃ©

\*/

public class DataAndTime {

public static void main(String[] args) {

List<Person> persons = new ArrayList<>();

try (

BufferedReader reader =

new BufferedReader(

new InputStreamReader(

DataAndTime.class.getResourceAsStream("people.txt")));

Stream<String> stream = reader.lines();

) {

stream.map(

line -> {

String[] s = line.split(" ");

String name = s[0].trim();

int year = Integer.parseInt(s[1]);

Month month = Month.of(Integer.parseInt(s[2]));

int day = Integer.parseInt(s[3]);

Person p = new Person(name, LocalDate.of(year, month, day));

persons.add(p);

return p;

})

.forEach(System.out::println);

} catch (IOException ioe) {

System.out.println(ioe);

}

LocalDate now = LocalDate.of(2014, Month.MARCH, 12);

persons.stream().forEach(

p -> {

Period period = Period.between(p.getDateOfBirth(), now);

System.out.println(p.getName() + " was born " +

period.get(ChronoUnit.YEARS) + " years and " +

period.get(ChronoUnit.MONTHS) + " months " +

"[" + p.getDateOfBirth().until(now, ChronoUnit.MONTHS)

+ " months]"

);

});

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

//package org.paumard;

**import** java.time.LocalDate;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** Person {

**private** String name;

**private** LocalDate dateOfBirth;

**public** Person(){}

**public** Person(String name, LocalDate dateOfBirth) {

**this**.name = name;

**this**.dateOfBirth = dateOfBirth;

}

**public** String getName() {

**return** name;

}

**public** LocalDate getDateOfBirth() {

**return** dateOfBirth;

}

@Override

**public** String toString() {

**return** "Person{" + "name=" + name + ", dateOfBirth=" + dateOfBirth + '}';

}

}

People.txt

Sarah 1999 12 15

Philip 1993 8 12

Beth 1991 6 5

Simon 1990 3 23

Nina 1991 7 12

Allan 1985 2 14

Leonard 1996 10 27

Barbara 1988 4 19

4 javafx

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.text.Font;

**import** javafx.stage.Stage;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** FirstApplication **extends** Application{

**public** **void** start(Stage stage) {

Label label = **new** Label("Hello world!");

label.setFont(**new** Font(50));

Scene scene = **new** Scene(label);

stage.setScene(scene);

stage.show();

}

**public** **static** **void** main(String[] args) {

*launch*();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

//package org.paumard.javafx;

**import** java.net.URL;

**import** java.util.ResourceBundle;

**import** javafx.event.ActionEvent;

**import** javafx.fxml.FXML;

**import** javafx.fxml.Initializable;

**import** javafx.scene.control.PasswordField;

**import** javafx.scene.control.TextField;

**import** sun.security.util.Password;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** MyController **implements** Initializable {

@FXML

**private** TextField username;

@FXML

**private** PasswordField password;

@Override

**public** **void** initialize(URL url, ResourceBundle bundle) {

}

**public** **void** okAction(ActionEvent event) {

System.***out***.println("Click ok button :" + username.getText());

}

**public** **void** cancelAction(ActionEvent event) {

System.***out***.println("Click cancel button :" + password.getText());

}

}

//package org.paumard.javafx;

**import** java.io.IOException;

**import** javafx.application.Application;

//import static javafx.application.Application.launch;

**import** javafx.fxml.FXMLLoader;

**import** javafx.scene.Parent;

**import** javafx.scene.Scene;

**import** javafx.stage.Stage;

/\*\*

\*

\* **@author** JosÃ©

\*/

**public** **class** SecondApplication **extends** Application {

@Override

**public** **void** start(Stage stage) {

**try** {

FXMLLoader loader =

**new** FXMLLoader(

getClass().getResource("ihm.fxml"));

Parent root = loader.load();

stage.setScene(**new** Scene(root));

stage.show();

} **catch** (IOException ioe) {

System.***out***.println(ioe.getMessage());

}

}

**public** **static** **void** main(String[] args) {

*launch*();

}

}

Ihm.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.\*?>

<?import javafx.scene.control.\*?>

<?import javafx.scene.layout.\*?>

<?import javafx.scene.paint.\*?>

<GridPane xmlns:fx="http://javafx.com/fxml"

fx:controller="org.paumard.javafx.MyController"

hgap="10" vgap="10">

<padding>

<Insets bottom="10.0" top="10.0"

left="10.0" right="10.0" />

</padding>

<children>

<Label text="User name:"

GridPane.columnIndex="0" GridPane.rowIndex="0"

GridPane.halignment="RIGHT"/>

<TextField fx:id="username"

GridPane.columnIndex="1" GridPane.rowIndex="0"/>

<Label text="Password:"

GridPane.columnIndex="0" GridPane.rowIndex="1"

GridPane.halignment="RIGHT"/>

<PasswordField fx:id="password"

GridPane.columnIndex="1" GridPane.rowIndex="1"/>

<HBox GridPane.columnIndex="0" GridPane.rowIndex="2"

GridPane.columnSpan="2" alignment="CENTER"

spacing="10">

<children>

<Button text="Ok" onAction="#okAction"/>

<Button text="Cancel" onAction="#cancelAction"/>

</children>

</HBox>

</children>

</GridPane>