Exercises on Collections

1. **Create a list of elements as shown below into the LinkedList and name it as languages.**

**C**

**C++**

**Java**

**Kotlin**

**Python**

**Perl**

**Ruby**

**Display the created list.**

**Remove an element at index 5 and display the list**

**Remove ‘Kotlin’ and display the list**

**Remove all the scripting languages (Python, Ruby, Perl) with one statement and display the list.**

**Remove all the elements from the list and display the list.**

**package** com.collections.pll;

**import** java.util.Iterator;

**import** java.util.LinkedList;

**import** java.util.List;

**public** **class** Exercise001 {

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

List<String> languages = **new** LinkedList<>();

languages.add("C");

languages.add("C++");

languages.add("Java");

languages.add("Kotlin");

languages.add("Python");

languages.add("Perl");

languages.add("Ruby");

//System.out.println("List of languages: " + languages);

//1.

System.***out***.println("List of languages");

**for**(String language : languages)

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

**for**(**int** i=0; i<languages.size(); i++)

System.***out***.println(languages.get(i));

Iterator<String> itr = languages.iterator();

**while**(itr.hasNext()) {

System.***out***.println(itr.next());

}

//2.Remove an element at index 5 and display the list

languages.remove(5);

System.***out***.println("List of languages after deleting the element at index 5");

**for**(String language : languages)

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

//3. Remove ‘Kotlin’ and display the list

languages.remove("Kotlin");

System.***out***.println("List of languages after deleting the element 'Kotlin'");

**for**(String language : languages)

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

//4. Remove all the scripting languages (Python, Ruby, Perl)

// with one statement and display the list.

List<String> removeList = **new** LinkedList<String>();

removeList.add("Python");

removeList.add("Ruby");

removeList.add("Perl");

languages.removeAll(removeList);

System.***out***.println("List of languages after deleting the elements 'Python, Ruby, Perl'");

**for**(String language : languages)

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

//5. remove all the list

languages.removeAll(languages);

System.***out***.println("List of languages after deleting all the elements.");

/\*

for(String language : languages)

System.out.println(language);

\*/

System.***out***.println("Elements after deleting all the elements: " + languages.size());

}

}

1. Create an ArrayList object students, and store the following employees in that list.

Student

studentcode

studentname

age

state

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Code** | **Student Name** | **Age** | **State** |
| AF0216223 | Aryan Raj | 21 | Andra Pradhesh |
| AF0216224 | Bivor Kumar | 22 | Andra Pradhesh |
| AF0216227 | SUSHMITA KUMARI | 23 | Madhya Pradesh |
| AF0216231 | Ragiv Zafar | 24 | Maharashtra |
| AF0216232 | RAHUL MAHTO | 25 | Orissa |
| AF0216234 | Nainsi Kumari | 19 | Gujarath |
| AF0216236 | MD ALI | 26 | Madhya Pradesh |
| AF0216238 | ABHISHEK KUMAR | 22 | Andra Pradhesh |
| AF0216240 | AFROZ ANSARI | 21 | Maharashtra |
| AF0216259 | RITIK RAJ | 20 | Orissa |
| AF0216263 | Anant Kumar | 25 | Andra Pradhesh |
| AF0216305 | BABU KUMAR | 24 | Gujarath |
| AF0216307 | NIKITA KUMARI | 23 | Madhya Pradesh |
| AF0216353 | PRANAV PANDEY | 22 | Orissa |
| AF0216964 | Radheshyam Kumar | 21 | Maharashtra |
| AF0216973 | Ankur Utpal | 19 | Gujarath |
| AF0217607 | Banty Mishra | 18 | Madhya Pradesh |
| AF0217615 | RIYA KUMARI | 20 | Madhya Pradesh |
| AF0217791 | Priyadarshani Kumari | 21 | Maharashtra |
| AF0223373 | Chanchal Thakur | 22 | Orissa |
| AF0221549 | SATENDRA KUMAR | 23 | Maharashtra |
| AF0216212 | Pappi Verma | 24 |  |
| AF0216244 | Ramkrishna Kushwah | 25 | Andra Pradhesh |
| AF0216245 | Yogita Tamoliya | 26 | Orissa |
| AF0216246 | Goutam Rathore | 20 | Maharashtra |
| AF0216247 | Bablu jadhav | 19 | Madhya Pradesh |
| AF0216248 | Arbaj Sheikh | 22 | Madhya Pradesh |
| AF0216256 | Raja Saini | 26 | Madhya Pradesh |
| AF0216284 | Divyanshu Tiwari | 24 | Gujarath |
| AF0217621 | Yash Upadhyay | 22 | Maharashtra |
| AF0090331 | JAGADISH CHATURVEDI | 20 | Maharashtra |
| AF0208998 | MISHAN CHANDRAKALA | 21 | Gujarath |
| AF0209009 | PAWAN PAWAR | 19 | Karnataka |
| AF0188217 | DIVYAM PANDEY | 28 | Goa |
| AF0210410 | AMRUTA PRAMODRAO PHUKE | 26 | Gujarath |
| AF0188227 | V DELHI RAJU | 24 | Goa |
| AF0213766 | ACHUGATLA ELIYAZ BASHA | 20 | Karnataka |
| AF0173587 | ANKIT KUMAR GOUD | 19 | Karnataka |
| AF0143570 | BHAGYASHRI RAMAKANT VISHWASRAO | 18 | Karnataka |
| AF0130214 | RAMAN SUMAN | 22 | Goa |
| AF0109212 | SOUGATA PRAMANIK | 22 | Gujarath |
| AF0149241 | SOUMYA RANJAN NAYAK | 22 | Karnataka |
| AF0188268 | SONU PARMAR | 25 | Goa |
| AF0208755 | M ARCHANA | 25 | Karnataka |
| AF0082432 | PRADEEP SHUKLA | 25 | Gujarath |

**Find the students aged over 20**

**Find students from the state Andhra Pradhesh**

**Sort employees by their age.**

**package** com.collections.bll;

**public** **class** Student **implements** Comparable<Student> {

**private** String studentCode;

**private** String studentName;

**private** Integer age;

**private** String state;

**public** Student() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Student(String studentCode, String studentName, Integer age, String state) {

**super**();

**this**.studentCode = studentCode;

**this**.studentName = studentName;

**this**.age = age;

**this**.state = state;

}

**public** String getStudentCode() {

**return** studentCode;

}

**public** **void** setStudentCode(String studentCode) {

**this**.studentCode = studentCode;

}

**public** String getStudentName() {

**return** studentName;

}

**public** **void** setStudentName(String studentName) {

**this**.studentName = studentName;

}

**public** Integer getAge() {

**return** age;

}

**public** **void** setAge(Integer age) {

**this**.age = age;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

@Override

**public** String toString() {

**return** "Student [studentCode=" + studentCode + ", studentName=" + studentName + ", age=" + age + ", state="

+ state + "]";

}

// Sort students by state and student name.

@Override

**public** **int** compareTo(Student student) {

/\*if(getState().compareTo(student.getState()) == 0)

return 0;

else if(getState().compareTo(student.getState()) > 0)

return 1;

else

return -1;

\*/

//return getState().compareTo(student.getState());

**if**(getState().compareTo(student.getState()) == 0)

**return** getStudentName().compareTo(student.getStudentName());

**else** **if**(getState().compareTo(student.getState()) > 0)

**return** getStudentName().compareTo(student.getStudentName());

**else**

**return** getStudentName().compareTo(student.getStudentName());

}

/\*

@Override

public int compareTo(Student student) {

if(getAge() == student.getAge())

return 0;

else if(getAge() > student.getAge())

return 1;

else

return -1;

}

\*/

}

**package** com.collections.pll;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**import** com.collections.bll.Student;

**public** **class** Exercise002 {

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

List<Student> studentList = **new** ArrayList<Student>();

studentList.add(**new** Student("AF0216223", "Aryan Raj", 21, "Andra Pradhesh"));

studentList.add(**new** Student("AF0216224", "Bivor Kumar", 25, "Andra Pradhesh"));

studentList.add(**new** Student("AF0216227", "SUSHMITA KUMARI", 23, "Madhya Pradesh"));

studentList.add(**new** Student("AF0216231", "Ragiv Zafar", 24, "Maharashtra"));

studentList.add(**new** Student("AF0216232", "RAHUL MAHTO", 25, "Orissa"));

studentList.add(**new** Student("AF0216234", "Nainsi Kumari", 19, "Gujarath"));

studentList.add(**new** Student("AF0216236", "MD ALI", 26, "Madhya Pradesh"));

studentList.add(**new** Student("AF0216238", "ABHISHEK KUMAR", 22, "Andra Pradhesh"));

studentList.add(**new** Student("AF0216240", "AFROZ ANSARI", 21, "Maharashtra"));

studentList.add(**new** Student("AF0216259", "RITIK RAJ", 20, "Orissa"));

studentList.add(**new** Student("AF0216263", "Anant Kumar", 25, "Andra Pradhesh"));

studentList.add(**new** Student("AF0216305", "BABU KUMAR", 24, "Gujarath"));

studentList.add(**new** Student("AF0216307", "NIKITA KUMARI", 23, "Madhya Pradesh"));

studentList.add(**new** Student("AF0216353", "PRANAV PANDEY", 22, "Orissa"));

studentList.add(**new** Student("AF0216964", "Radheshyam Kumar", 21, "Maharashtra"));

studentList.add(**new** Student("AF0216973", "Ankur Utpal", 19, "Gujarath"));

studentList.add(**new** Student("AF0217607", "Banty Mishra", 18, "Madhya Pradesh"));

studentList.add(**new** Student("AF0217615", "RIYA KUMARI", 20, "Karnataka"));

studentList.add(**new** Student("AF0217791", "Priyadarshani Kumari", 21, "Maharashtra"));

studentList.add(**new** Student("AF0223373", "Chanchal Thakur", 22, "Orissa"));

studentList.add(**new** Student("AF0221549", "SATENDRA KUMAR", 23, "Maharashtra"));

studentList.add(**new** Student("AF0216212", "Pappi Verma", 24, "Kerala"));

studentList.add(**new** Student("AF0216244", "Ramkrishna Kushwah", 25, "Andra Pradhesh"));

studentList.add(**new** Student("AF0216245", "Yogita Tamoliya", 26, "Orissa"));

// 1. Find the students aged over 20

**for**(Student student:studentList)

**if**(student.getAge() > 20)

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

List<Student> greateThan20 = **new** ArrayList<Student>();

**for**(Student student:studentList)

**if**(student.getAge() > 20)

greateThan20.add(student);

System.***out***.println("Students greater thab 20 age: " + greateThan20);

**for**(Student student:greateThan20)

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

// 2. Find students from the state Andhra Pradhesh

System.***out***.println("Students from the state Andhra Pradhesh\n\n");

**for**(Student student:studentList) {

**if**(student.getState().equals("Andra Pradhesh")) {

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

}

}

//3. Sort students by their age.

Collections.*sort*(studentList);

**for**(Student student:studentList) {

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

}

//4. Sort students by state and student name.

Collections.*sort*(studentList);

**for**(Student student:studentList) {

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

}

}

}

1. **Given an element write a program to check if element (value) exists in ArrayList?**
2. **Write a program to add elements to the HashMap given the key and value data type is String?**
3. Write a static method to find the sum of all the even numbers in an ArrayList. Within main, create a list with at least 10 integers and call your method on the list.

public static int sumEven(ArrayList<Integer> arr) {

int total = 0;

for (int integer : arr) {

if (integer % 2 == 0) {

total += integer;

}

}

return total;

}

1. Write a static method to print out each word in a list that has exactly 5 letters.
2. Modify your code to prompt the user to enter the word length for the search.
3. Make a program  StudentHashMap that does the following:
4. It takes in student names and ID numbers (as integers) instead of names and grades.
5. The keys should be the IDs and the values should be the names.
6. Display the values.
7. Make a **Map** that associates the following employee IDs with names. Keys and values of Maps can be any Object type, so in real life you would probably have the key be a String and the associated value be a Person or Employee object.

To make things simpler on this exercise, you can use String for both the ID and the name, rather than bothering to create a Person or Employee class.

The point here is to associate keys with values, then retrieve values later based on keys.

|  |  |
| --- | --- |
| ID | Name |
| a1234 | Steve Jobs |
| a1235 | Scott McNealy |
| a1236 | Jeff Bezos |
| a1237 | Larry Ellison |
| a1238 | Bill Gates |

To test several valid and invalid ID’s and print the associated name.

1. Create a class Person. A person has a first name and a last name.

Write a main-program that creates an ArrayList with a number of people (e.g. Priscilla Wagner, Tom Parker, Elvis Presley).

Sort the arraylist based on the first name and show the result:

Elvis Presley

Priscilla Wagner

Tom Parker

Extra: when a class implements the Comparable interface it can be sorted without a comparator: Collections.sort(people).

Can you change the Person class so that it will be sorted according to last name?

public static void main(String[] args) {

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

people.add(new Person("Priscilla", "Wagner"));

people.add(new Person("Tom", "Parker"));

people.add(new Person("Elvis", "Presley"));

Collections.sort(people, new Comparator<Person>() {

@Override

public int compare(Person o1, Person o2) {

return o1.getFirstName().compareTo(o2.getFirstName());

}

});

for(Person p: people){

System.out.println(p);

}

}

public class Person implements Comparable<Person>{

private String firstName;

private String lastName;

public Person(String firstName, String lastName) {

super();

this.firstName = firstName;

this.lastName = lastName;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

@Override

public String toString() {

return String.format("%s %s",firstName, lastName);

}

@Override

public int compareTo(Person arg0) {

// TODO Auto-generated method stub

return 0;

}

}

1. Write a program that reads in a series of first names and stores them in a LinkedList.

Do not store duplicate names.

Allow the user to search for a first name.

1. Write a program that reads a file and displays the words of that file as a list.

Then display them in reverse order.

Then display them with all plurals (ending in "s") capitalized.

Then display them with all plural words removed.

// Removes all plural words from the given list.

public static void removePlural(ArrayList<String> list) {

for (int i = 0; i < list.size(); i++) {

String str = list.get(i);

if (str.endsWith("s")) {

list.remove(i);

i--;

}

}

}

1. Write a program that reads a file full of numbers and

displays all the numbers as a list, then:

– Prints the average of the numbers.

– Prints the highest and lowest number.

– Filters out all of the even numbers (ones divisible by 2).