JavaClass28 Maps in Java

Maps in Java:

Maps are datastructures that allow us to store key value pairs.

Why we should use Maps?

Whenever we have to store key value pairs instead of using arrays or 2D arrays or ArrayLists we should use

Maps because they provide very good performance and have built-in methods to process the data.

What is the syntax of Maps?

Map<DataTypeOfKey,DataTypeOfValue> anyName=new MapClass<>();

for example if we have to store fruit and their prices we will do something like this

HashMap<String,Double> fruit=new HashMap<>();

Types Of Maps:

1) HashMap 2) LinkedHashMap 3) TreeMap

HashMap:

This is the most widely used type of map reason, it is fastest and occupies less memory.

we should always use Hashmap if there are no requirements for insertion order or sorting.

LinkedHashMap:

LinkedHashMap keeps the insertion order, but it takes a little more memory than HashMap and is a little slower.

TreeMap:

TreeMap sorts the data in natural order but insertion is very slow. it takes more memory to keep the data in sorted

order.

FileHandling:

Reading writing data to files with the help of java is called file handling.

Steps to read data from any files on any computer:

1)location of file on our computer.

2) we need to go to that location.

3) we need a software that can read and write data to that file for exmaple we need Microsoft Excel if we want to work with xlsx files.

package class28;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileInputStream;

import java.io.IOException;

public class ExcelDemo1 {

public static void main(String[] args) throws IOException {

// we need to know path of the file

String path="/Users/apple/IdeaProjects/SDETBatch14Java/Data/Test.xlsx";

//navigate to this path

FileInputStream fileInputStream=new FileInputStream(path);

// that special that can handle the xlsx files

XSSFWorkbook excelFile=new XSSFWorkbook(fileInputStream);

//Getting to the Sheet inside the Excel file where data is stored

Sheet sheet=excelFile.getSheet("Sheet1");

// getting the Row that contains the data rows are index based, so they will start from zero

Row row0=sheet.getRow(0);

// get the cell that contains the data cells are also index based

System.out.println(row0.getCell(0));

// get the data from row 1

Row row1=sheet.getRow(1);

System.out.println(row1.getCell(0));

}

}

package class28;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileInputStream;

import java.io.IOException;

public class ExcelDemo2 {

public static void main(String[] args) throws IOException {

String path="/Users/apple/IdeaProjects/SDETBatch14Java/Data/Test.xlsx";

FileInputStream fileInputStream=new FileInputStream(path);

XSSFWorkbook excel=new XSSFWorkbook(fileInputStream);

Sheet sheet=excel.getSheet("Sheet1");

int noOfRows= sheet.getPhysicalNumberOfRows(); // returns how many rows in actual contains the data in sheet

System.out.println(noOfRows);

for (int i = 0; i < noOfRows; i++) {

Row row= sheet.getRow(i);

int noOfCell=row.getPhysicalNumberOfCells(); // returns the total number of cells that contain the data

for (int j = 0; j <noOfCell ; j++) {

Cell cell=row.getCell(j);

System.out.print(cell+" ");

}

System.out.println();

}

}

}

package class28;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.Properties;

public class FileDemo1 {

public static void main(String[] args) throws IOException {

/\*

rightclick on your file and click copypathreference then click from Content root

\*/

var path="/Users/apple/IdeaProjects/SDETBatch14Java/Data/config.properties"; //location of the file

var fileInputStream=new FileInputStream(path); //it helps us Navigate to the file

var properties=new Properties(); // that special software which helps us read data from that file

properties.load(fileInputStream); //loads all the data from the file inside(Memory)

System.out.println(properties.getProperty("URl"));

System.out.println(properties.getProperty("password"));

fileInputStream.close(); // closes the file

// Break till 1:145

}

}

package class28;

import java.util.HashMap;

import java.util.Set;

public class MapDemo3 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

Set<String> keys =items.keySet(); // keySet method returns a Set that contains all the keys of a map

// Collection<String> keys2 =items.keySet();

for (String key:keys

) {

System.out.println(key);

}

//Break till 11:50

// write code to print all the values from a map using a loop

}

}

package class28;

import java.util.HashMap;

public class MapDemo4 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

for (Double value: items.values()

) {

System.out.println(value);

}

}

}

package class28;

import java.util.HashMap;

public class MapDemo5 {

// var lastname; does not work with instance variables as java can't figure out the data type as there are no values

// assigned

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

//Set<Map.Entry<String,Double>> entrySet =items.entrySet();

var entrySet=items.entrySet();

/\* var dog=new Dog();

var number=10.2;\*/

System.out.println(entrySet);

for (var item:entrySet

) {

System.out.println(item);

}

}

}

package class28;

import java.util.HashMap;

public class MapDemo6 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

/\* var iterator=items.entrySet().iterator();

while (iterator.hasNext()){

var item=iterator.next(); //gets the items

var key=item.getKey();

var value=item.getValue();

if(key.contains("e")&& value>10){

iterator.remove();

}

}

\*/

items.entrySet().removeIf(entry-> entry.getKey().contains("e")&&entry.getValue()>10);

System.out.println(items);

}

}

package class28;

import java.util.HashMap;

public class MapsDemo {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

//i want to remove the items if their price is more than 10

//first we are getting a collection of values then an Iterator

/\* Iterator<Double> iterator=items.values().iterator(); //first we are getting a collection of values then an Iterator

while (iterator.hasNext()){

Double price=iterator.next();

if(price>10){

iterator.remove();

}

}\*/

System.out.println(items);

items.values().removeIf(price -> price>10);

}

/\*public boolean check(Double price){

if (price > 10) {

return true;

} else {

return false;

}

}\*/

}

package class28;

import java.util.HashMap;

public class MapsDemo2 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

items.keySet().removeIf(name-> name.contains("e"));

System.out.println(items);

}

}

package class28;

import java.util.HashMap;

import java.util.Set;

public class MapDemo3 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

Set<String> keys =items.keySet(); // keySet method returns a Set that contains all the keys of a map

// Collection<String> keys2 =items.keySet();

for (String key:keys

) {

System.out.println(key);

}

//Break till 11:50

// write code to print all the values from a map using a loop

}

}

package class28;

import java.util.HashMap;

public class MapDemo4 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

for (Double value: items.values()

) {

System.out.println(value);

}

}

}

package class28;

import java.util.HashMap;

public class MapDemo5 {

// var lastname; does not work with instance variables as java can't figure out the data type as there are no values

// assigned

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

//Set<Map.Entry<String,Double>> entrySet =items.entrySet();

var entrySet=items.entrySet();

/\* var dog=new Dog();

var number=10.2;\*/

System.out.println(entrySet);

for (var item:entrySet

) {

System.out.println(item);

}

}

}

package class28;

import java.util.HashMap;

public class MapDemo6 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

/\* var iterator=items.entrySet().iterator();

while (iterator.hasNext()){

var item=iterator.next(); //gets the items

var key=item.getKey();

var value=item.getValue();

if(key.contains("e")&& value>10){

iterator.remove();

}

}

\*/

items.entrySet().removeIf(entry-> entry.getKey().contains("e")&&entry.getValue()>10);

System.out.println(items);

}

}

package class28;

import java.util.HashMap;

public class MapsDemo {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

//i want to remove the items if their price is more than 10

//first we are getting a collection of values then an Iterator

/\* Iterator<Double> iterator=items.values().iterator(); //first we are getting a collection of values then an Iterator

while (iterator.hasNext()){

Double price=iterator.next();

if(price>10){

iterator.remove();

}

}\*/

System.out.println(items);

items.values().removeIf(price -> price>10);

}

/\*public boolean check(Double price){

if (price > 10) {

return true;

} else {

return false;

}

}\*/

}

package class28;

import java.util.HashMap;

public class MapsDemo2 {

public static void main(String[] args) {

HashMap<String,Double> items=new HashMap<>();

items.put("Iphone 14",1000.0);

items.put("Eggs",2.40);

items.put("Apples",5.0);

items.put("Keyboard",100.0);

items.put("Flower",1.0);

items.keySet().removeIf(name-> name.contains("e"));

System.out.println(items);

}

}