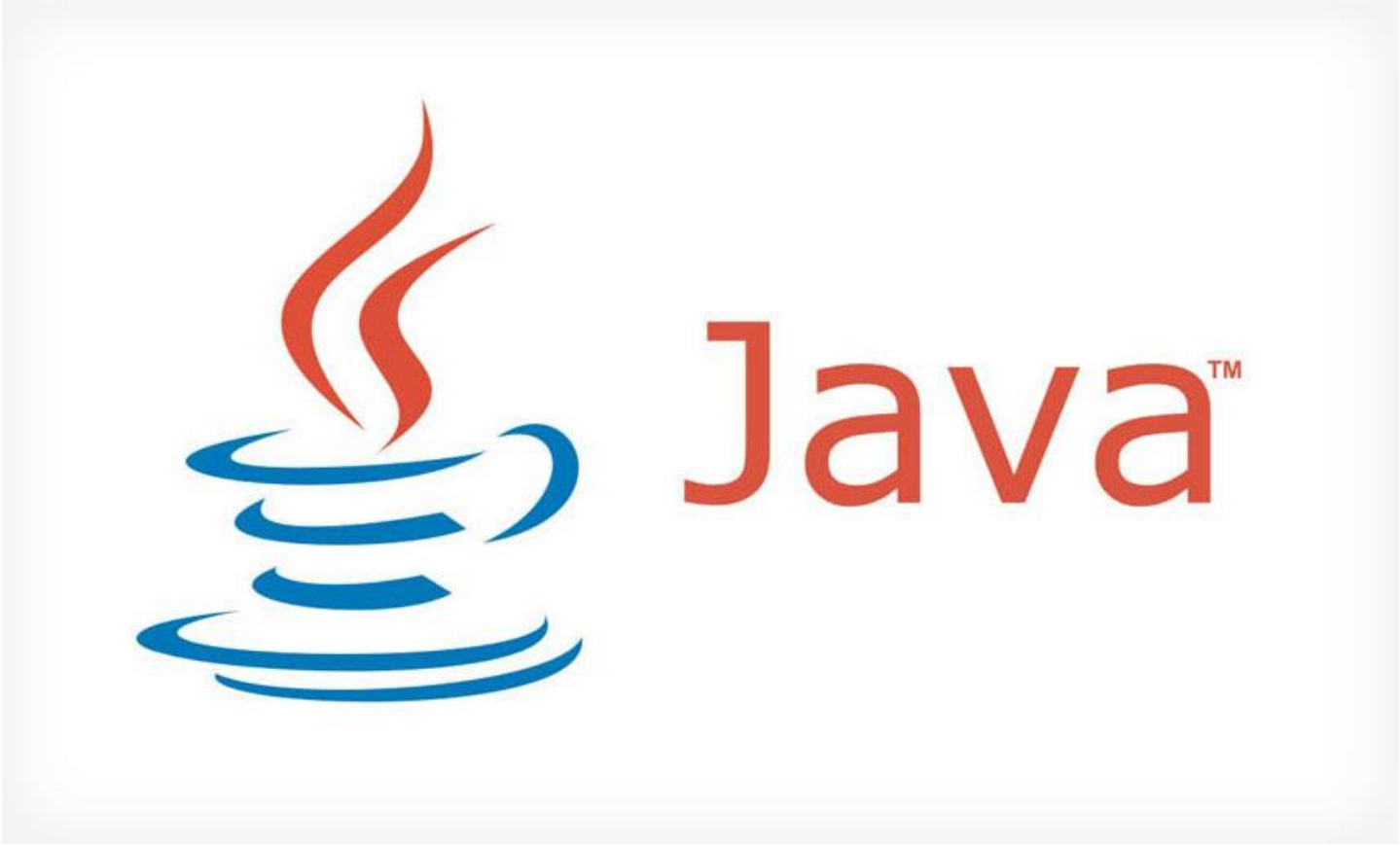
وظيفة JAVA



تقديم الطلاب :

محمد الشيخ عمر

زكريا الشيخ

ميرفت المرة

ربى عبد القادر

اشراف المهندس :مضر عباس

Class Drive

Name	ViewDrives
description	تابع لطباعة الأقراص الموجودة على الحاسب وتابع الـlistRoots يرد مصفوفة من نمط *file* تحتوي على الأقراص الموجودة

Method	`public void ViewDrives(){`
	`  File[] roots = File.listRoots();`
	`  System.out.println("AllDirves\t\tTotalSpace");`
	`  for (File root : roots) {`
	`  System.out.println(root + "\t\t" + root.getTotalSpace()`
	`  + " bytes");`
	`  }`
	`  }`

Name	ViewDirve
description	تابع لطباعة محتويات قرص ما يأخذ وسيط وهو اسم القرص شرح لبعض التوابع الجاهزة المستخدمة ***File exists*** وهو تابع يرد قيمة منطقية في حال وجود الملف يرد *True* خلاف ذلك يرد *False* ***File listFiles*** وهو تابع يرد مصفوفة من نوع *File* كل عنصر من هذه المصفوفة هو مؤشر على ملف او مجلد داخل القرص ***File getName*** وهو تابع يأخذ وسيط من نوع ملف ويرد اسم لهذا الملف **getSizeFolder** وهذا التابع يرد *Long* حجم الملف او المجلد يأخذ وسيط من نوع ملف وهو تابع عودي لحساب ما اذا كان يوجد مجلد داخل مجلد

Method	`public void ViewDirve(String Name){`
	`  File file = new File(Name + ":\\");`
	`  if(file.exists()){`
	`  File[] files = file.listFiles();`
	`  if(files != null){`
	`  for (File file1 : files) {`
	`  System.out.println(file1.getName() + "\t"`
	`  + getSizeFolder(file1));`
	`  }`
	`  }`
	`  }`
	`  }`

Name	getSizeFolder
description	وهذا التابع يرد *Long* حجم الملف او المجلد يأخد وسيط من نوع ملف وهو تابع عودي لحساب ما اذا كان يوجد مجلد داخل مجلد ***File length*** هذا التابع يرد حجم الملف فقط وليس المجلد لذلك نختبر اذا كان الملف الي نتعامل معه ملف ام مجلد ***File isFile*** يرد قيمة منطقية في حال كان الملف الذي نتعامل معه ملف ام لا

Method	public long getSizeFolder(File dir) {
	long size = 0;
	for (File file : dir.listFiles()) {
	if (file.isFile()) {
	size += file.length();
	}
	else
	size += getSizeFolder(file);
	}
	return size;
	}

Class Folder

Name	CreatNewFolder
description	تابع لأنشاء مجلد باستخدام التابع mkdir() يأخذ وسيط وهو مسار المجلد المراد انشائه
Method	

```
public void CreatNewFolder(String path){
      File file  =  new File(path);
      if(! file. exists()){
         file. mkdir();
         System. out. println("The Folder Is Created ^_^");
      }
      else{
         System. out. println("The Folder Is Already Exists!! ");
      }
   }
```

Name	DeleteFolder
description	تابع لحذف مجلد يأخذ وسيط وهو مسار المجلد المراد حذفه نسخدم هنا تابع deleteDirectory وهو تابع عودي لحدف المجلد
Method	

```
public void DeleteFolder(String path){
      File file  =  new File(path);
      if(file. exists()){
         deleteDirectory(file);
         System. out. println("The Folder Is Deleted");
      }
      else{
         System. out. println("The Folder Is Not Exists!! ");
      }
   }
```

Name	deleteDirectory
description	تابع لحذف مجلد يرد قيمة منطقية في حال حذفه ام لا يأخذ وسيط وهو الملف المراد حذفه وهو تابع عودي نختبر اذا كان الملف isDirectory نستدعي التابع من اجله ليقوم بحذف كافة الملفات التي بداخلة لأنه لايمكننا حذف مجلد بشكل كامل إلا لنقوم بحذف محتويات هذا المجلد **File isDirectory** تابع يرد قيمة منطقية في حال كان *file* مجلد يرد *True* خلاف ذلك اي ان *file* يدل على ملف فيرد *False* **File delete**

تابع يقوم بحذف ملف فقط

Method

```
public boolean deleteDirectory(File directory) {
    if(directory.exists()){
      File[] files = directory.listFiles();
      if(files != null){
        for (File file : files) {
          if (file.isDirectory()) {
            deleteDirectory(file);
          } else {
            file.delete();
          }
        }
      }
    }
    return(directory.delete());
  }
```

Name	**SearchFolder**
description	تابع للبحث عن مجلد يرد قيمة منطقية ويأخذ وسيط وهو مسار الملف

Method

```
public boolean SearchFolder(String path){
    File file = new File(path);
    if(file.exists()){
      File[] files = file.listFiles();
      if(files != null){
        for (File file1 : files) {
          if (file1.isDirectory()) {
            deleteDirectory(file1);
          } else {
            return false;
          }
        }
      }
    }
    return false;
  }
```

SortByName	*Name*
تابع لترتيب العناصر حسب الاسم اذا كان العنصر موجود نقوم بانشاء *list* واضافة العناصر عليها وبعدها نقوم باستدعاء تابع الـsort الذي يقوم بترتيب العناصر	***description***

public void SortByName(String path) {
File file = new File(path);
if(file.exists()){
String [] ss ;
ss = file.list();
List < String > list = new ArrayList <> ();
list.addAll(Arrays.asList(ss));
Collections.sort(list);
for (String list1 : list) {
System.out.println(list1);
}
}
else
System.out.println("The Folder IsNot Exists");
}
	Method

SortBySize	*Name*
تابع لترتيب العناصر حسب الحجم اذا كان العنصر موجود نقوم بانشاء mapو نقوم بالمرور على جميع عناصر الملف اذا كان مجلد نضع حجمه واسمه بالـmap والا نضع طول الملف واسمه وبعدها نقوم بانشاء *set* وهي عبارة عن مجموعة غير مرتبة وتحتوي على عناصر مكررة والـ TreeSet يقوم بتخزين العناصر في شجرة	***description***

public void SortBySize(String path) {
File file = new File(path);
Map < Long, String > map = new HashMap <> ();
if(file.exists()){
File[] files = file.listFiles();
if(files ! = null){
for (File file1 : files) {
if (file1.isDirectory()) {
map.put(getSizeFolder(file1), file1.getName());
} else {
map.put(file1.length(), file1.getName());
}
}
Set < Long > keys = map.keySet();
TreeSet < Long > sortedkeys = new TreeSet <> (keys);
for(Long key : sortedkeys){
System.out.println(map.get(key));
}
}
}
else
System.out.println("The Folder IsNot Exists!! ");
}
	Method

Class MyFile

CreatNewFile	Name
تابع لأنشاء ملف وسيطه مسار الملف **createNewFile** هو تابع يرد قيمة منطقية في حال انشائه يرد True عدا ذلك يرد False	description

public void CreatNewFile(String path) throws IOException {
File file = new File(path);
if(file.createNewFile()){
System.out.println("The File Is Created");
}
else{
System.out.println("The File Is Already Exists!! ");
}
}
	Method

DeleteFile	Name
تابع لحذف ملف وسيطه هو مسار الملف	description

public void DeleteFile(String path){
File file = new File(path);
if(file.exists()){
file.delete();
System.out.println("The File Is Deleted");
}
else{
System.out.println("The File Is Not Exists!!");
}
}
	Mathod

SearchFile	Name
تابع للبحث عن ملف يرد قيمة منطقية في حال ايجاده يرد True وفي حال عدم ايجاده يرد False	description

public boolean SearchFile(String path){
File file = new File(path);
return file.exists();
}
	Method

بنى المعطيات:

ArrayList: هي مصفوفة يمكن تغير حجمها بالـ Run Time ويكبر حجمها عند إضافة عنصر لها وعدد عناصرها يساوي سعتها ولقدد استخدمنا في الوظيفة الـ ArrayList بدلا من الـ vector لان الـ ArrayList ليست Sychroniz لذلك فهي افضل .

Set : هي مجموعة مرتبة تحتوي على عناصر غير مكررة وهي عبارة عن interface ولها ابن وهو الـ TreeSet الذي يخزن العناصر في شجرة ثنائية .

Map: وهي مجموعة من البنى تخزن المعطيات بطريقة Key-Value وهي عبارة عن interface و للإضافة فيها نستخدم التابع put .