Заремба Анастасия

М18-Ш03

17 билет

*Вопрос:*

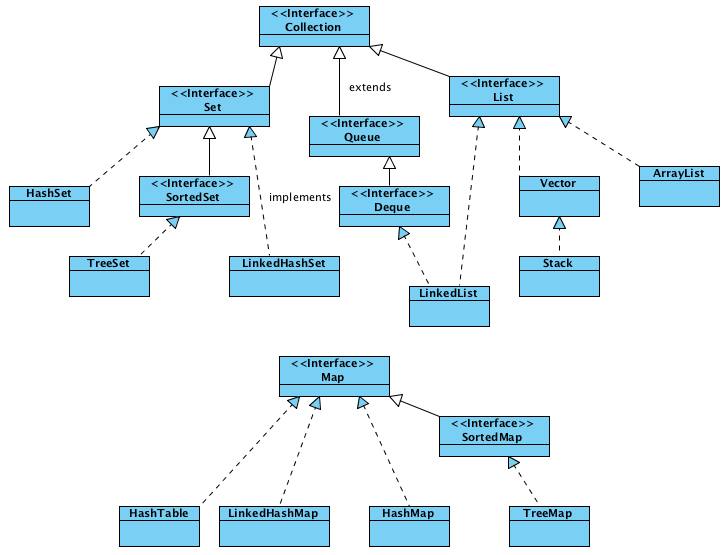
*Что вы знаете о коллекциях типа List? Как синхронизировать метод? Что вы знаете об интерфейсе FilenameFilter?*

Ответ:

Коллекциями/контейнерами в Java принято называть классы, основная цель которых – хранить набор других элементов.

Массивы обладают значительными недостатками. Одним из них является конечный размер массива, как следствие, необходимость следить за размером массива. Другим — индексная адресация, что не всегда удобно, т.к. ограничивает возможности добавления и удаления объектов. Чтобы избавиться от этих недостатков уже несколько десятилетий программисты используют **рекурсивные типы данных**, такие как **списки** и **деревья**. Стандартный набор коллекций Java служит для избавления программиста от необходимости самостоятельно реализовывать эти типы данных и снабжает его дополнительными возможностями.

Коллекция имеет иерархию, представленную на рисунке ниже



List – это упорядоченный список. Объекты хранятся в порядке их добавления в список. Доступ к элементам списка осуществляется по индексу.

Интерфейс java.io.FilenameFilter может быть реализован для фильтрации имен файлов в определенной папке. Интерфейс FilenameFilter содержит метод boolean accept (File dir, String name). Класс должен реализовывать этот метод, а каждый тестируемый файл должен быть включен в общий список файлов.

Интерфейс FilenameFilter позволяет создавать объекты, которые фильтруют списки файлов и удаляют из них ненужные. Он содержит всего один метод: boolean accept(File dir, String name).

Возвращает true, если файл с именем name в каталоге dir должен входить в отфильтрованный список.

Например, можно реализовать проверку всех файлов содержащихся в папке и отобразить только те которые имеют нужное расширение.

package ua.com.prologistic;  
  
import java.io.File;  
import java.io.FilenameFilter;  
  
public class FileNameFilterExample {  
  
public static void main(String[] args) { // будем искать в папке tmp  
String dir = "/Users/prologistic/tmp"; // в этой папке будем искать файлы с расширением .xml  
String ext = ".xml"; // вызываем метод поиска файлов с расширением .xml в папке tmp  
findFiles(dir, ext);  
}  
  
// метод поиска  
private static void findFiles(String dir, String ext) {  
File file = new File(dir);  
if(!file.exists()) System.out.println(dir + " папка не существует");  
File[] listFiles = file.listFiles(new MyFileNameFilter(ext));  
if(listFiles.length == 0){  
System.out.println(dir + " не содержит файлов с расширением " + ext);  
}else{  
for(File f : listFiles)  
System.out.println("Файл: " + dir + File.separator + f.getName());  
}  
}  
  
// Реализация интерфейса FileNameFilter  
public static class MyFileNameFilter implements FilenameFilter{  
  
private String ext;  
  
public MyFileNameFilter(String ext){  
this.ext = ext.toLowerCase();  
}  
[@Override](https://vk.com/id384774802)  
public boolean accept(File dir, String name) {  
return name.toLowerCase().endsWith(ext);  
}  
}  
}

Класс MyFileNameFilter реализует интерфейс FilenameFilter. Метод accept проверяет имя файла на наличие определенного расширения. В методе main() мы вызываем метод findFiles, который использует MyFileNameFilter для определения файлов в формате XML.

levf. dct