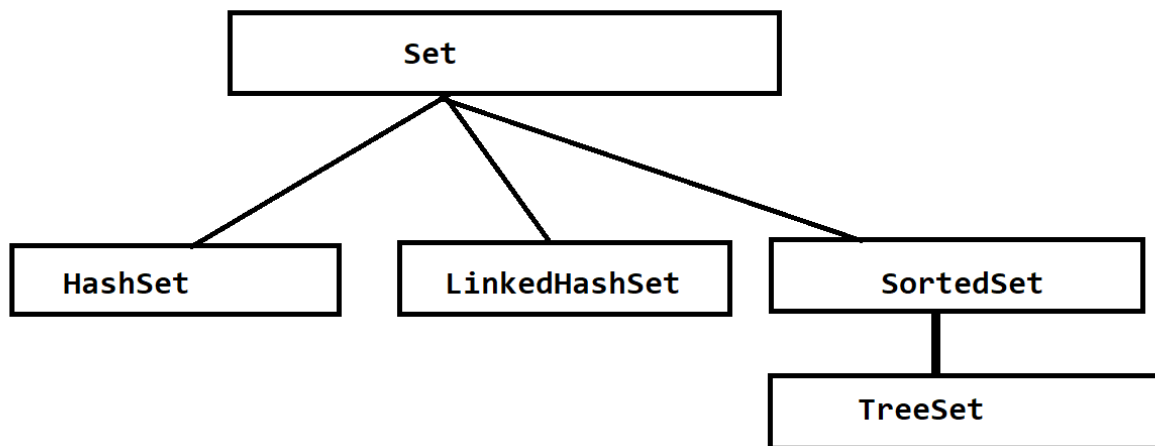


Set Collection

Set Collection which is used for store the unique values and store the data by using hashing technique called as Set Collection Set cannot maintain the index for element.

Three Types of Set Collection

Following Diagram shows the hierarchy of Set collection



1) HashSet: HashSet store the unique elements and Generate the element sequence randomly

2) LinkedHashSet: LinkedHashSet store the unique values and generate the element sequence according to the user sequence.

3) TreeSet: TreeSet is used for store the unique values and generate the all values in ascending order.

Now we want to create the application to perform basic operation on Set collection using HashSet

1. Add Element in HashSet
 2. View All element from HashSet
 3. Search Element in HashSet
 4. remove element from HashSet
 5. Count the all element of HashSet
- Etc

Example of HashSet

```

package org.techhub;
import java.util.*;
public class HashSetApplication {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HashSet hs = new HashSet();
        do {
            Scanner xyz = new Scanner(System.in);
            System.out.println("1:Add Element in HashSet");
            System.out.println("2:View All Element from HashSet");
            System.out.println("3:Search Element in HashSet");
            System.out.println("4:Remove element from HashSet");
            System.out.println("5:Count the element from HashSet");
            System.out.println("Enter your choice");
            int choice = xyz.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("Enter the value in set");
                    int val = xyz.nextInt();
                    hs.add(val);
                    break;
                case 2:
                    Iterator i = hs.iterator();
                    while (i.hasNext()) {
                        Object obj = i.next();
                        System.out.println(obj);
                    }
                    break;
                case 3:
                    System.out.println("Enter the search value");
                    int sval = xyz.nextInt();
                    boolean b = hs.contains(sval);
                    if (b) {
                        System.out.println("Element found");
                    } else {
                        System.out.println("element not found");
                    }
            }
        }
    }
}

```

```

        break;
    case 4:
        System.out.println("Enter value for delete");
        val = xyz.nextInt();
        b = hs.remove(val);
        if (b) {
            System.out.println("Value removed success...");
        } else {
            System.out.println("Value not removed");
        }
        break;
    case 5:
        System.out.println("Number of element in collection " + hs.size());
        break;
    default:
        System.out.println("Wrong choice");
    }

} while (true); // infinite loop

}

}

```

Example of LinkedHashSet

```

package org.techhub;
import java.util.*;
public class HashSetApplication {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        LinkedHashSet hs = new LinkedHashSet();
        do {
            Scanner xyz = new Scanner(System.in);
            System.out.println("1:Add Element in HashSet");
            System.out.println("2:View All Element from HashSet");
            System.out.println("3:Search Element in HashSet");
            System.out.println("4:Remove element from HashSet");

```

```
System.out.println("5:Count the element from HashSet");
System.out.println("Enter your choice");
int choice = xyz.nextInt();
switch (choice) {
case 1:
    System.out.println("Enter the value in set");
    int val = xyz.nextInt();
    hs.add(val);
    break;
case 2:
    Iterator i = hs.iterator();
    while (i.hasNext()) {
        Object obj = i.next();
        System.out.println(obj);
    }
    break;
case 3:
    System.out.println("Enter the search value");
    int sval = xyz.nextInt();
    boolean b = hs.contains(sval);
    if (b) {
        System.out.println("Element found");
    } else {
        System.out.println("element not found");
    }
    break;
case 4:
    System.out.println("Enter value for delete");
    val = xyz.nextInt();
    b = hs.remove(val);
    if (b) {
        System.out.println("Value removed success...");
    } else {
        System.out.println("Value not removed");
    }
    break;
case 5:
```

```

        System.out.println("Number of element in collection " + hs.size());
        break;
    default:
        System.out.println("Wrong choice");

    }

} while (true); // infinite loop

}

}

```

Example of TreeSet Collection

```

package org.techhub;
import java.util.*;
public class HashSetApplication {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        TreeSet hs = new TreeSet();
        do {
            Scanner xyz = new Scanner(System.in);
            System.out.println("1:Add Element in HashSet");
            System.out.println("2:View All Element from HashSet");
            System.out.println("3:Search Element in HashSet");
            System.out.println("4:Remove element from HashSet");
            System.out.println("5:Count the element from HashSet");
            System.out.println("Enter your choice");
            int choice = xyz.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("Enter the value in set");
                    int val = xyz.nextInt();
                    hs.add(val);
                    break;
                case 2:
                    Iterator i = hs.iterator();
                    while (i.hasNext()) {

```

```

        Object obj = i.next();
        System.out.println(obj);
    }
    break;
case 3:
    System.out.println("Enter the search value");
    int sval = xyz.nextInt();
    boolean b = hs.contains(sval);
    if (b) {
        System.out.println("Element found");
    } else {
        System.out.println("element not found");
    }
    break;
case 4:
    System.out.println("Enter value for delete");
    val = xyz.nextInt();
    b = hs.remove(val);
    if (b) {
        System.out.println("Value removed success...");
    } else {
        System.out.println("Value not removed");
    }
    break;
case 5:
    System.out.println("Number of element in collection " + hs.size());
    break;
default:
    System.out.println("Wrong choice");
}
} while (true); // infinite loop
}
}

```

How To Arrange data in descending order using TreeSet

If we want to arrange data in descending order using TreeSet we have the

descendingSet() method of TreeSet and this method return the reference of NavigableSet interface and NavigableSet interface is used for fetch data of Set using descending order.

```
package org.techhub;
import java.util.*;
public class TreeSetApplication {

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        TreeSet ts = new TreeSet();
        ts.add(4);
        ts.add(1);
        ts.add(20);
        ts.add(33);
        ts.add(2);
        ts.add(5);
        NavigableSet nav=ts.descendingSet();
        Iterator i=nav.iterator();
        while(i.hasNext())
        { Object obj = i.next();
          System.out.println(obj);
        }
    }
}
```

Mini Project Application By using HashSet

WAP to create Bag with ball collection and we want to add the balls in bag and we want extract the single ball from back but we want to extract the ball randomly and we want to show the information with ball color, company ,weight ,size

1) Create the POJO class for store the ball information

```
package org.techhub.ball;
```

```

public class Ball {

    private String color;
    public String getColor() {
        return color;
    }
    public void setColor(String color) {
        this.color = color;
    }
    public int getWeight() {
        return weight;
    }

    public void setWeight(int weight) {
        this.weight = weight;
    }
    public int getSize() {
        return size;
    }
    public void setSize(int size) {
        this.size = size;
    }
    public String getCompany() {
        return company;
    }
    public void setCompany(String company) {
        this.company = company;
    }

    private int weight;
    private int size;
    private String company;
}

```

2) create the class name as BallExtraction and declaration the void addBall() method in it and add the new ball in HashSet collection


```

package org.techhub.ball;
import java.util.*;
public class BallExtraction {
    HashSet bag = new HashSet();

    public void addBall(Ball ball) {
        bag.add(ball);
    }
}

```

3) create the class with main method and in main method we have to create the object of BallExtraction first and then we have to create the object of Ball and store the information in ball object and pass to ball object to BallExtraction for store in collection.

```

package org.techhub.ball;
import java.util.*;
public class BallExtractionApplication {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BallExtraction be = new BallExtraction();
        do {
            Scanner xyz = new Scanner(System.in);
            int choice;
            System.out.println("1:Add Ball in Bag");
            System.out.println("2:View All Balls in bag");
            System.out.println("3:Extract the ball from bag");
            System.out.println("enter your choice");
            choice = xyz.nextInt();
            switch (choice) {
                case 1:
                    Ball ball = new Ball();
                    xyz.nextLine();
                    System.out.println("enter the ball color");

```

```

        String colorName = xyz.nextLine();
        System.out.println("enter the ball company");
        String compName = xyz.nextLine();
        System.out.println("Enter the ball weight");
        int weight = xyz.nextInt();
        System.out.println("Enter the ball size");
        int size = xyz.nextInt();
        ball.setColor(colorName);
        ball.setCompany(compName);
        ball.setWeight(weight);
        ball.setSize(size);
        be.addBall(ball);
        break;
    case 2:
        break;
    case 3:
        break;
    default:
        System.out.println("Wrong choice");
    }
} while (true);

}

}

```

4) create the method viewAllBalls() in BallExtraction and fetch all data from collection and display it

BallExtraction.java

```

package org.techhub.ball;

import java.util.*;

public class BallExtraction {
    HashSet bag = new HashSet();

```

```

    public void addBall(Ball ball) {
        bag.add(ball);
    }
    public void viewAllBalls()
    {
        Iterator i=bag.iterator();
        while(i.hasNext())
        {
            Object obj = i.next();
            Ball b =(Ball)obj;
            System.out.println(b.getColor()+"\t"+b.getCompany()+"\t"+b.getSize()
+"\\t"+b.getWeight());
        }
    }
}

```

BallExtractionApplication.java

```

package org.techhub.ball;

import java.util.*;

public class BallExtractionApplication {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BallExtraction be = new BallExtraction();
        do {
            Scanner xyz = new Scanner(System.in);
            int choice;
            System.out.println("1:Add Ball in Bag");
            System.out.println("2:View All Balls in bag");
            System.out.println("3:Extract the ball from bag");

```

```

        System.out.println("enter your choice");
        choice = xyz.nextInt();
        switch (choice) {
            case 1:
                Ball ball = new Ball();
                xyz.nextLine();
                System.out.println("enter the ball color");
                String colorName = xyz.nextLine();
                System.out.println("enter the ball company");
                String compName = xyz.nextLine();
                System.out.println("Enter the ball weight");
                int weight = xyz.nextInt();
                System.out.println("Enter the ball size");
                int size = xyz.nextInt();
                ball.setColor(colorName);
                ball.setCompany(compName);
                ball.setWeight(weight);
                ball.setSize(size);
                be.addBall(ball);
                break;
            case 2:
                be.viewAllBalls();
                break;
            case 3:
                break;
            default:
                System.out.println("Wrong choice");
        }
    } while (true);
}

```

5) Write the method in BallExtraction class name as pickUpBall() and fetch ball from collection randomly

BallExtraction.java

package org.techhub.ball;
import java.util.*;
public class BallExtraction {
 HashSet **bag** = **new** HashSet();

 public void addBall(Ball **ball**) {
 bag.add(**ball**);
 }

 public void viewAllBalls()
 {
 Iterator **i**=**bag**.iterator();
 while(**i.hasNext**())
 {
 Object **obj** = **i.next**();
 Ball **b**=(Ball)**obj**;

 System.**out**.println(**b.getColor**()+"\t"+**b.getCompany**()+"\t"+**b.getSize**()
 +"\t"+**b.getWeight**());
 }
 }

 public void pickUpBall()
 {

 Iterator **i**=**bag**.iterator();
 if(**i.hasNext**())
 {
 Object **obj** = **i.next**();
 Ball **b**=(Ball)**obj**;
 System.**out**.println(**b.getColor**()+"\t"+**b.getCompany**());
 i.remove();

 }
 else
 {

 System.**out**.println("You bag is empty no ball found");

```

    }
}

```

BallExtractionApplication.java

```
package org.techhub.ball;
```

```
import java.util.*;
```

```
public class BallExtractionApplication {
```

```

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BallExtraction be = new BallExtraction();
        do {
            Scanner xyz = new Scanner(System.in);
            int choice;
            System.out.println("1:Add Ball in Bag");
            System.out.println("2:View All Balls in bag");
            System.out.println("3:Extract the ball from bag");
            System.out.println("enter your choice");
            choice = xyz.nextInt();
            switch (choice) {
                case 1:
                    Ball ball = new Ball();
                    xyz.nextLine();
                    System.out.println("enter the ball color");
                    String colorName = xyz.nextLine();
                    System.out.println("enter the ball company");
                    String compName = xyz.nextLine();
                    System.out.println("Enter the ball weight");
                    int weight = xyz.nextInt();
                    System.out.println("Enter the ball size");
                    int size = xyz.nextInt();
                    ball.setColor(colorName);
                    ball.setCompany(compName);
                    ball.setWeight(weight);
                    ball.setSize(size);

```

```
        be.addBall(ball);
        break;
    case 2:
        be.viewAllBalls();
        break;
    case 3:
        be.pickUpBall();
        break;
    default:
        System.out.println("Wrong choice");
    }
} while (true);
}
}
```