```
package alfa1;
import java.io.IOException;
import java.io.PrintWriter;
import java.nio.charset.Charset;
import static java.nio.charset.StandardCharsets.UTF_8;
import java.nio.file.Files;
import static java.nio.file.Files.lines;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Arrays;
import java.util.List;
//import java.StringBuilder;

public class Alfa1 {

public static void main(String[] args) {
//competition area
int x1=-1,x2=-1,y1=-1,y2=-1;
//start finish area
int x3=-1,x4=-1,y3=-1,y4=-1;
//checkpoint area
int x5=-1,x6=-1,y5=-1,y6=-1;
int n = 0,i=0,j=0,o=0;

int [] m = new int[10];
double[] h=new double[10] ;
String[] boat_identifier = new String[20];
String [] boat_name = new String[20];

int [] crew_number=new int[10];
int [][] crew_identifier= new int [20][20];
int [][] captain=new int [20][20];
int [][] crew_years_sailing= new int [20][20];
String [][] crew_first_name= new String [20][20];
String [][] crew_last_name =new String [20][20];
int [] nr_lines= new int [20];
int [][] boat_x = new int [20][20];
// int [] nr_y= new int [20];
int [][] boat_y = new int [20][20];
// int [] nr_time= new int [20];
int [][] boat_time= new int [20][20];
try {
List<String> allLines =
Files.readAllLines(Paths.get("C:\\Users\\danut\\Desktop\\Alfa1\\in&out\\t
est1.in"));

String[] strings = allLines.stream().toArray(String[]::new);
// reading  the coordinates of competition area
String[] splitStr = strings[0].split("\\s+");

try {
x1 = Integer.parseInt(splitStr[0]);
y1= Integer.parseInt(splitStr[1]);
x2 = Integer.parseInt(splitStr[2]);
y2= Integer.parseInt(splitStr[3]);
} catch (NumberFormatException e) {
System.out.println("Invalid coordinates for competition area");
}
// reading the coordinates of Start/Finish area
```

```
splitStr = strings[1].split("\\s+");

try {
x3 = Integer.parseInt(splitStr[0]);
y3= Integer.parseInt(splitStr[1]);
x4 = Integer.parseInt(splitStr[2]);
y4= Integer.parseInt(splitStr[3]);
} catch (NumberFormatException e) {
System.out.println("Invalid coordinates for start/finish area");
}

// reading  the coordinates of Checkpoint area
splitStr = strings[2].split("\\s+");

try {
x5 = Integer.parseInt(splitStr[0]);
y5= Integer.parseInt(splitStr[1]);
x6 = Integer.parseInt(splitStr[2]);
y6= Integer.parseInt(splitStr[3]);
} catch (NumberFormatException e) {
System.out.println("Invalid coordinates for checkpoint area ");
}

try {
// reading  the number of boats
n= Integer.parseInt(strings[3]);
} catch (NumberFormatException e) {
System.out.println("Invalid data for the number of boats ");
}
//reading  the boats and crew menbers
int nr=4;
for(i=0;i<n;i++)
{

splitStr = strings[nr].split("\\s+");

try {
boat_identifier[i]=splitStr[0];
boat_name[i]=splitStr[1];
h[i]=Double.parseDouble(splitStr[2]);
crew_number[i]=Integer.parseInt(splitStr[3]);
}
catch (NumberFormatException e) {
System.out.println("Invalid data for boat number "+ i+e);
}
// gathering data for each crew member for a specific boat
for(j=0;j<crew_number[i];j++)
{
nr++;
splitStr = strings[nr].split("\\s+");

try {
crew_identifier[i][j]=Integer.parseInt(splitStr[0]);
crew_first_name[i][j]=splitStr[1];
crew_last_name[i][j]=splitStr[2];
captain[i][j]=Integer.parseInt(splitStr[3]);
crew_years_sailing[i][j]=Integer.parseInt(splitStr[4]);

} catch (NumberFormatException e) {
System.out.println("Invalid data for crew members in the boat "+i );
```

```
}

}
nr++;

}
for(i=0;i<n;i++)
{
nr_lines[i]=0;

}
String id;

while(nr<strings.length)
{
splitStr = strings[nr].split("\\s+");
id=splitStr[0];

o=0;
// assuming that are in a random order

while(o<n)
{
if(id.equals(boat_identifier[o]))
{


    boat_x[o][nr_lines[o]]=Integer.parseInt(splitStr[1]);
    boat_y[o][nr_lines[o]]=Integer.parseInt(splitStr[2]);
    boat_time[o][nr_lines[o]]=Integer.parseInt(splitStr[3]);
    nr_lines[o]++;

    o=n;

}
o++;

}
nr++;

}


// checking the boats
int nr_comp=0, nr_finish=0,nr_check=0;
int [] nr_boat_finish= new int [20];
int [] nr_boat_checkpoint= new int [20];
int [] contor =new int [10];
int [] total_boat_times= new int [10];
StringBuilder sb = new StringBuilder();
StringBuilder partial_standings = new StringBuilder();
StringBuilder final_standings = new StringBuilder();
// for finish are neccesary 2 croses of the start finsih area
for(i=0;i<n;i++)
{
for(j=0;j<nr_lines[i];j++)
{//competition area  coordonates checking and counting
if(boat_x[i][j]<x1 || boat_x[i][j]>x2 )
    nr_comp++;
if(boat_y[i][j]<y1 || boat_y[i][j]>y2 )
```

```
    nr_comp++;



// start / finish area checking  and counting the number of passes
(requred 2 )
if((boat_x[i][j]>x3 && boat_x[i][j]<x4) && (boat_y[i][j]>y3 &&
boat_y[i][j]<y4))
    {
    nr_boat_finish[i]++;
    }

// checkpoint area checking

if((boat_x[i][j]>x5 && boat_x[i][j]<x6) && (boat_y[i][j]>y5 &&
boat_y[i][j]<y6))
    {
    nr_boat_checkpoint[i]++;
    }

total_boat_times[i]=total_boat_times[i]+boat_time[i][j];

}
total_boat_times[i]=total_boat_times[i]/60;

if(nr_boat_finish[i]<2)
{nr_finish++;
sb.append(boat_name[i]+ " ");

}
if(nr_boat_checkpoint[i]<1)
{nr_check++;
sb.append(boat_name[i]+ " ");

}
// System.out.println(nr_boat_finish[i] + " "+nr_boat_checkpoint[i] );
if(nr_boat_finish[i]==2 && nr_boat_checkpoint[i]==1)

partial_standings.append(boat_name[i]+ "( "+total_boat_times[i] +
"minutes),  ");
final_standings.append(boat_name[i]+ "( "+total_boat_times[i]*h[i]+ "
minutes/ "+h[i]+ "), " );


}

StringBuilder captains = new StringBuilder();
for(i=0;i<n;i++)
for(j=0;j<crew_number[i];j++)
if(captain[i][j]==1 && crew_years_sailing[i][j]>2)
{
captains.append(crew_first_name[i][j]+ " "+ crew_last_name[i][j] + " ("+
crew_years_sailing[i][j]+ " years ) ");
}

// results :)
```

```
//String x ="Number of boats outside the competition area: "+nr_comp;
List<String> lines = Arrays.asList("Number of boats outside the
competition area: "+nr_comp);
Path file =
Paths.get("C:\\Users\\danut\\Desktop\\Alfa1\\in&out\\test_alfa.out");
//Files.write(file,lines, Charset.forName("UTF-8") );


Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);

lines=Arrays.asList("Number of boats that didn’t enter the finish area:
"+nr_finish);
Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);
lines=Arrays.asList("Number of boats that didn’t enter the checkpoint
area: "+ nr_check);
Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);
// System.out.println("Number of boats that didn’t enter the finish area:
"+nr_finish);
//System.out.println("Number of boats that didn’t enter the checkpoint
area: "+ nr_check);
//System.out.println(x1+" "+ x2+" "+x2);
//System.out.println("Number of boats outside the competition area:
"+nr_comp);
// String names = null;

lines=Arrays.asList("Boats that didn’t finish the race: "+sb);
Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);

lines=Arrays.asList("Most experienced captains: "+captains);
Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);

lines=Arrays.asList("Partial standings: : "+partial_standings);
Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);
lines=Arrays.asList("Final standings: : "+final_standings);
Files.write(file, (lines +
System.lineSeparator()).getBytes(UTF_8),StandardOpenOption.C
REATE,StandardOpenOption.APPEND);
System.out.println("  all good");


} catch (IOException e) {
e.printStackTrace();
}
//System.out.println(x1);


}
```

```
}
```