"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XIX – 2016 – Issue 2
The journal is indexed in: PROQUEST / DOAJ / Crossref / EBSCOhost / INDEX COPERNICUS / DRJI / OAJI / JOURNAL INDEX / I2OR / SCIENCE LIBRARY INDEX / Google Scholar / Academic Keys/ ROAD Open Access / Academic Resources / Scientific Indexing Services / SCIPIO / JIFACTOR

USING JAVA PROGRAMMING LANGUAGE TO CODIFY/DECODIFY HYDROMETEOROLOGICAL MESSAGES FROM SHIP CODE ABOARD VESSELS AND WRITING OF PARAMETERS USING BJERKNESS SCHEME

[1]Dinu ATODIRESEI
[2]Andrei BAUTU
[3]Alecu TOMA
[4]Andrei POCORA
[5]Ionut BRANICI

[1]Adv.Instructor,PhD, Department of Navigation and Naval Transport,"Mircea cel Batran" Naval Academy, Romania,dinu.atodiresei@anmb.ro
[2]Lecturer,PhD, Department of Navigation and Naval Transport,"Mircea cel Batran" Naval Academy, Romania, andrei.bautu@anmb.ro
[3]Adv.Instructor,PhD, Department of Navigation and Naval Transport,"Mircea cel Batran" Naval Academy, Romania,alecu.toma@anmb.ro
[4]Instructor,Eng.,PhD Candidate Department of Navigation and Naval Transport,"Mircea cel Batran" Naval Academy,Romania,andrei.pocora@anmb.ro
[5]Eng.,Maritime Officer Constanta, Romania

Abstract: *With the help of Java programming language, a software for conversion of hydrometeorological parameters received by vessels, buoys and shore stationsin SHIP code. These informations are further plotted around the shore station using Bjerknes scheme. The software can also be used the other way around, in encoding own measurements and transmitting them to regional forecast stations, according to requirements implied by shipping companies or Naval Forces.*

Keywords: *meteorological data analysis,Java program,weather charts, shipstation circle, Bjerknes scheme.*

1. Introduction

Weather forecast is very important aboard commercial vessels during voyages or in case of military scenarios aboard navy ships. There are multiple software developed by companies such as Weather Routing Inc, Seaware, AMI, Weathernews Int., C-Map, SMHI, SPOS, that offer daily assistance in route optimizing [1].

Problems occur when hydrometeorological informaton is sent encoded by specialized vessels and meteorological shore stations, and officers aboard vessels are obliged to decode them (most often in SHIP code) or when they have to code own information (the case of shipping companies fom North America).Encoding and decoding hydrometeorological messages can be complicated and requires a lot of time and personell. Thus, the developed software involves processing encoded hydrometeorological information, measured and sent by surface vessels, buoys, meteorological shore stations in SHIP code, as IMO (International Maritime Organization) requires, with the following general formula:BBXXYYGGi$_W$99LaLaLaQLoLoLoLoi$_R$i$_X$h VVNddff1s$_n$TTT2s$_s$T$_d$T$_d$T$_d$4PPPP5appp6RRRt$_R$7 wwW$_1$W$_2$8N$_n$C$_L$C$_M$C$_H$222D$_S$V$_S$0s$_n$T$_W$T$_W$T$_W$2P$_W$P$_W$H$_W$H$_W$3d$_{w1}$d$_{w1}$d$_{w2}$d$_{w2}$4P$_{W1}$P$_{W1}$H$_{W1}$H$_{W1}$5P$_{W2}$P$_{W2}$ H$_{W2}$H$_{W2}$6I$_S$E$_S$E$_S$R$_S$ICEc$_j$S$_j$b$_j$D$_j$Z$_j$[2]. The representation around a maritime weather forecast station can be realised as follows:(Figure1)[3].

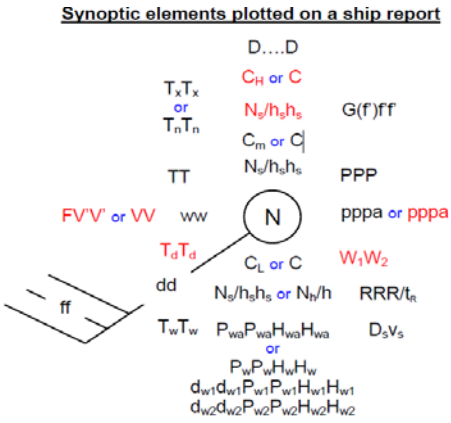

Fig.1. Hydrometeorological parameters received in SHIP code and their representation around a maritime weather forecast station(buoy, shore station, research vessel) using Bjerknes scheme.

DOI: 10.21279/1454-864X-16-I2-002

Also, with the software, the transformation of information measured by meteorological equipment aboard vessels into SHIP code can be achieved in a very short time.

2. Description of algorithm and software

2.1 Java Software

Java programming language is an object-oriented language designed by Sun Microsystems in the early 90s.The language allows writting in a unitary manner, programs for different platforms, from servers and desktops to embeded devices.Basically, Java programming language provides a uniform platform to the programmer, a platform that abstracts the hardware on which it runs.

According to[4][5], the programming language lends a great part of the syntax from C and C++. A properly written Java program can run without modification on any platform that is installed on a Java Virtual Machine (JVM), because sources are compiled into instructions for Java JVM, independent of the hardware platform used. Because JVM specification is public there are a lot of JVM developers such as Oracle, IBM, Bea, FSF.

2.2 Encoding hydrometeorological information in Java language program

For this software,*Scanner* objects have been used to input numbers from keypad, and to take each number, or group of numbers, in order to create the final code. The input data will be entered from the keyboard as text (*String* objects). In order to multiply it, the data must be transformed into *Integer*, using perseInt function:

intvar=Integer.parseInt(str);

With the help of *System.out.println*command, the user receives a message asking for input values from the keypad in order to encode parameters of weather forecast.For example, wind indicator can be: 0,1, 3 sau 4.

```
System.out.println("Indicatorulpentruvanteste:");
System.out.println("0-vantestimatinm/s;1-
vantmasuratcuanemometrulinm/s;);");
System.out.println("3-vantestimatinNd;4-
vantmasuratcuanemometrulinNd;);");
```

These lines of code will display the following message in the console: (Figure 2):

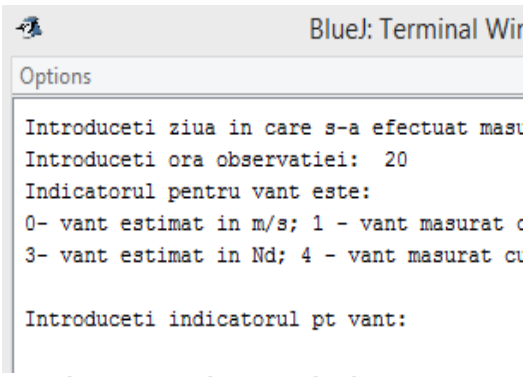

Fig.2 Execution terminal

The same steps were repeated or every group of numbers from SHIP code, thus helping the user to make the right choice in the concsole.

This next example represents the code lines necessary to encode the height of clouds. (Figure 3).

```
System.out.println("Tastati : 0- daca baza plafonului de nori este sub 50 m");
System.out.println("Tastati : 1 - daca baza plafonului de nori este intre 50-100m");
System.out.println("Tastati : 2 - daca baza plafonului de nori este intre 100-200 m");
System.out.println("Tastati : 3 - daca baza plafonului de nori este intre 200-300 m");
System.out.println("Tastati : 4 - daca baza plafonului de nori este intre 300-600 m");
System.out.println("Tastati : 5 - daca baza plafonului de nori este intre  600-1000 m");
System.out.println("Tastati : 6 - daca baza plafonului de nori este intre 1000-1500 m");
System.out.println("Tastati : 7 - daca baza plafonului de nori este intre 1500-2000 m");
System.out.println("Tastati : 8 - daca baza plafonului de nori este intre 2000-2500 m");
System.out.println("Tastati : 9 - daca nu sunt nori sub 2500 m ");
System.out.println(" ");
System.out.print("Introduceti valoarea la care se afla baza plafonului de nori: ");
```

Fig. 3 The code lines necessary to encode the height of clouds

At the end of the program all input from the user is gathered into one code, as follows:

-SHIP code groups are made up of 5 digits;

-each group is special, being formed out of 1,2,3,4 or 5 groups of digits entered separately from the keyboard;sometimes the first numbers out of a specific group represent the group indicative which is automatically written by the program.

For example, the first group as $YYGGi_W$will consist of three groups entered from the keyboard: the first group represents the day in which YY forecast was made, the second group represents the hour at which wind speed was measured *GG,* and the last digit represents the instrument used for measurements and if the speed was measured in m/s or Kn. After the three groups are entered, the program will join them as follows:

$YY*1000+GG*10+i_W$

If the number of the day or hours is smaller than 10, then the generated code can be incomplete.

DOI: 10.21279/1454-864X-16-I2-002

For example, if the reading was made on the 2nd day of a month, the program will display an incomplete code:2*1000+GG*10+i_W=2GGi_W,thus the group will have 4 digits instead of 5. In order to avoid such errors, *if* instruction was used (Figure 4).

```
if(v8==0 & w8<=9)
 System.out.print(" 00"+w8*100+x8);
else if(v8==0 & w8>=9)
 System.out.print(" 0"+w8*100+x8);
if(v8!=0 & w8<=9)
 System.out.print(" "+v8+"0"+w8*100+x8);
else
{
 System.out.print(" " + (v8*10000+w8*100+x8))
}//gr 5
```

Fig. 4 Java code for the display of group 8

In the previous code sequence, v8 represents the input value for encoding nebulosity, while the second image presents the condition that the value must fulfill: if the sky is clear v8 must be equal to zero, so the program was given instructions to automatically write zero and then continue to do multiplication and assemblies.
The next sequence of Java code presents the encoding of the first 6 groups:

```
/* Grupa 1 */
System.out.print(v2*1000+v3*10+v4);
/* Grupa 2 */
System.out.print("99"+v5);
/* Grupa 3 */
System.out.print(""+(v6*10000+w6));
/* Grupa 4 */
if(x7==0)
System.out.print(""+v7*1000+w7*100+"0"+y7);
else
System.out.print(""+(v7*10000+w7*1000+x7*100+
y7));
/* Grupa 5 */
if(v8==0&w8<=9)
System.out.print("00"+w8*100+x8);
elseif(v8==0&w8>=9)
System.out.print("0"+w8*100+x8);
if(v8!=0&w8<=9)
System.out.print(""+v8+"0"+w8*100+x8);
else
System.out.print(""+(v8*10000+w8*100+x8));
/* Grupa 6 */
if(v9==0)
System.out.print("10"+w9);
else
System.out.print("1"+(v9*1000+w9));
```

As a result of this calculation, the software will display SHIP code according to the input values entered by the user.

2.3. Decoding hydrometeorological informations

Decoding of the received weather messagein SHIP code begins with the reading of each group from the code as a 5-digit integer, using the following steps:
1.initializing of an integer variable (denoted i) with 0;
2.using *for* instruction to read and save at most 19 groups (the maximum number of groups in SHIP code);
3.using a *Scanner*object, the software will save as terms all the groups entered by the user as SHIP code (Figure 5);

```
Scanner nr = new Scanner(System.in);

String valoare ;
System.out.print("Gr."+i+"= ");
valoare = nr.next();
int v = Integer.parseInt(valoare);
System.out.println(" ");
```

Fig. 5 Java code for reading the group of codes

Aftef the values are entered, the program wll decode user input using the following algorithm:
1. Checking the position of the term within the vector;
2. Dividing each group into subgroups, each signifying something together (the reverse process of codification, dividing by 1000, 100 or 10 depending on the group);
3. Using various decisions decoding will be made and informaton will be displayed.
For example, the software will execute the following sequence for decoding information regarding swell waves.

```
if(i==16)
{
if(v/100%100!=0)
System.out.println("Perioadaprimuluisistemdevalu
ridehulăestede"+v/100%100+"sec");
else
System.out.println("Nusuntsistemedevaluridehulă"
);

switch(v%100){
case0:
System.out.println("Înălțimeaprimuluisistemdevalu
ridehulăestede0.25m");
break;
case1:
```


DOI: 10.21279/1454-864X-16-I2-002

```
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsaintre0.25si0.75m");
break;
case2:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsaintre0.75si1.25m");
break;
case3:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsaintre1.25si1.75m");
break;
case4:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsaintre1.75si2.25m");
break;
case5:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsaintre2.25si2.75m");
break;
case6:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsaintre2.75si3.25m");
break;
case7:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre3.25si3.75m");
break;
case8:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre3.75si4.25m");
break;
case9:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre4.25si4.75m");
break;
case10:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre5.25si5.75m");
break;
case11:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre6.25si6.75m");
break;
case12:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre6.75si7.25m");
break;
case13:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre7.25si7.75m");
break;
case14:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre7.75si8.25m");
break;
case15:
System.out.println("Înălțimeaprimuluisistemdevaluridehulăestecuprinsăintre8.25si8.75m");
break;
default:
System.out.println("Nusuntvaluridehulă");
}
```

In a similar way decoding for the other 17 groups was made.

3. Results

3.1 Graphical representation of hydrometeorological information through Bjerknes's scheme using Java

Using Java language, an applet was created (a window of execution) where Bjerknes’s scheme is plotted automatically.

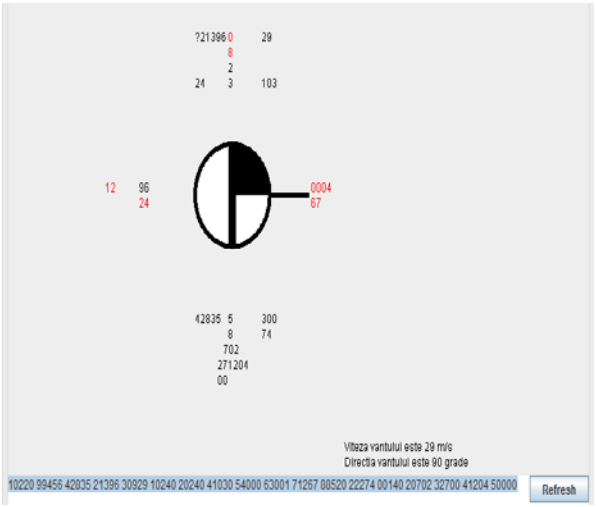

Fig.6. Bjerknes’s scheme applet

In order to obtain thes representation, the following Java libraries were used:

-java.awt.BasicStroke;
-java.awt.Color;
-java.awt.Dimension;
-java.awt.Font;
-java.awt.FontMetrics;
-java.awt.GradientPaint;
-java.awt.Graphics;
-java.awt.Graphics2D;
-java.awt.RenderingHints;
-java.awt.event.WindowAdapter;
-java.awt.event.WindowEvent;
-java.awt.geom.Arc2D;
-java.awt.geom.Ellipse2D;
-java.awt.geom.GeneralPath;
-java.awt.geom.Line2D;
-java.awt.geom.Rectangle2D;
-java.awt.geom.RoundRectangle2D;
-java.applet.Applet;
-javax.swing.JApplet;
-javax.swing.JFrame;
-java.awt.*;

DOI: 10.21279/1454-864X-16-I2-002

-java.applet.*;
-java.awt.*;
-java.awt.event.*;
-java.applet.*;
-javax.swing.*;
-java.util.*;

In the beginning of the program font, dimension of letters and numbers, dimensions and colour of curved and straight lines to be drawn in the applet were set, using the following lines of code

Graphics2Dg2=(Graphics2D)g;
g2.setPaint(Color.black);
g2.setStroke(newBasicStroke(5.0f));

Once these choices are made, a *v* variable, that saves the values of the nebulosity, will determine the filling of the circle symbol for nebulosity.For nebulosity 6/8 the graph is shown in figure 7 and for nebulosity 5/8 the graph is shown in figure 8.

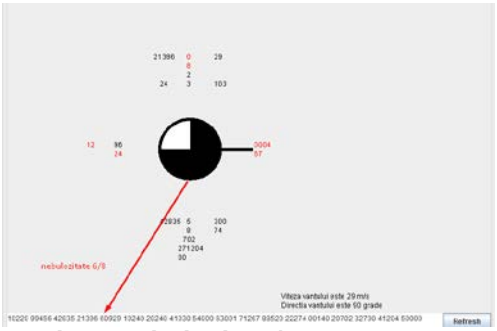

Fig.7. Nebulosity 6/8

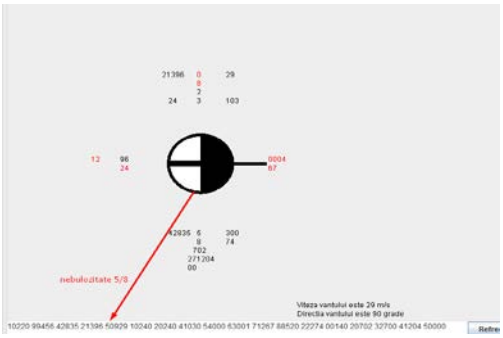

Fig.8 Nebulosity5/8

In order to display text on the graphics, *drawString*function was used and to change the color or the writing *setColor*function was used with parameters *Color.BLACK*and *Color.RED* as needed.

The line that shows the direction of the wind required calculus for the point of origin for the representation and the ending point of the vector. After calculus is done *g.drawLine(x_1,y_1, x_2, y_2)* function was used, where x_1,y_1 represents the origin point and x_2,$y_2$represents the ending point of the vector. The software will automatically choose what lines to draw according to the direction of the wind.

In the lower right part of the applet *Refresh* button was added to update the graphic when parameters change.

CONCLUSIONS

With the help of the developed software, coding/decoding SHIP code messages is easiser and faster. Also, their display takes less time, because it is not necessary to write them on a computer or by hand – messages are already in electronic format.As a perspective, this software can be improved for use with the help of an interface, data recorded by any meteorological equipment aboard vessels to achieve local weather forecasts.

BIBLIOGRAPHY

[1] C.U.BÖTTNER:*Weatherroutingforshipsindegradedcondition,*PublishedinReasearchGate,2016:https://www.researchgate.net/publication/250759038
[2] ***Ship's Code and Decode Book, Tenth Edition, 1981
[3] ***www.metoffice.gov.uk/NationalMeteorologicalLibraryandArchive
[4] TănasăȘ.,OlaruC.,StefanA.,2011*Java–dela0laexpert"*,**ISBN:**9789734624058EdituraPolirom,2011,864p
[5] ***,Java programming language on Wikipedia,https://en.wikipedia.org/wiki/Java_(programming_language)

DOI: 10.21279/1454-864X-16-I2-002