

# ПАРАМЕТРИ ОТ КОМАНДНИЯ РЕД ПРИ СОФТУЕР С ТЕКСТОВ ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС

Research described in this tutorial was partially supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science.

Developed by Velbazhd Software LLC

В основната си част съвременният софтуер е оформен с графичен потребителски интерфейс. Преди появата на графичния интерфейс диалогът с потребителя е воден в текстов режим с команден ред. Тази комуникация с команди прави софтуера значително по-труден за използване и е една от основните причини за появата на графичните среди.



Текстовият потребителски интерфейс намира приложение и в наши дни, особено когато става въпрос за сървърни приложения, които рядко изискват интензивна работа с крайния потребител.



Честа практика е софтуерът с текстов потребителски интерфейс да получава множество аргументи при стартирането на изпълнимия файл. Съвременните програмни езици дават достатъчно възможности входните аргументи да бъдат ефективно разчетени и използвани.





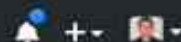
При софтуерни решения за пакетна обработка, параметрите за настройка се подават при стартирането на програмата и вследствие на своята работа програмата генерира съответните резултати. Такъв е случаят и с приложението за трансформация на пълноцветни растерни изображения към векторно представяне с редуцирани цветове.





Search or jump to...

Pull requests Issues Marketplace Explore



TodorBalabanov / Ellipses-Image-Approximator

Unwatch

1

Unstar

3

Fork

0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

It is an attempt to approximate full color image with limited color ellipses.

Edit

Manage topics

70 commits

1 branch

0 packages

0 releases

1 contributor

GPL-3.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



TodorBalabanov Many fixes of not working code were done.

Latest commit 6856937 yesterday

doc	Sorting by colors was wrong and it was fixed.	2 days ago
gradle/wrapper	Gradle build folders reorganization done.	11 days ago
input	Sorting by colors was wrong and it was fixed.	2 days ago
output	Read me file was updated.	yesterday
src	Many fixes of not working code were done.	yesterday
.gitignore	G Code template was added.	10 days ago
.travis.yml	Add file for automated building.	4 years ago
LICENSE	Initial commit	4 years ago
README.md	Better number of ellipses estimation was taken.	yesterday
build.gradle	Gradle build folders reorganization done.	11 days ago
gradlew	Gradle build folders reorganization done.	11 days ago
gradlew.bat	Gradle build folders reorganization done.	11 days ago
settings.gradle	Source code formatting was done.	11 days ago

README.md



## Ellipses Image Approximator

От командния ред към главната функция на програмата в Java се подава масив от символни низове, който съдържа параметрите при стартирането на виртуалната машина и приложението в нея.



```

62     run ant colony optimization.
63     */
64     for (int i = 0; i < Util.NUMBER_OF_REPETITIONS; i++) {
65         graph.resetTau();
66         AntColony4EIA colony = new AntColony4EIA(graph, Util.NUMBER_OF_ANTS,
67             Util.NUMBER_OF_ITERATIONS);
68         colony.start();
69     }
70 }
71
72 /**
73  * Single entry point of the program.
74  *
75  * @param args
76  *         Command line parameters.
77  * @throws Exception
78  *         Exception program stop.
79  */
80 public static void main(String[] args) throws Exception {
81     File input = new File(args[0]);
82     File output = new File(args[1]);
83     String path = output.getCanonicalPath() + "/";
84
85     original = ImageIO.read(input);
86
87     Ellipse.WIDTH = Integer.valueOf(args[4]);
88     Ellipse.HEIGHT = Integer.valueOf(args[5]);
89
90     colors.clear();
91     for (int i = 6; i < args.length; i++) {
92         colors.add(new Color(
93             Integer.parseInt(args[i], 16) | Util.ELLIPSES_ALPHA << 24,
94             true));
95     }
96
97     // TODO Should be some kind of external parameter. The area of the image
98     // divided by the area of the bounding rectangle of the simple graphic
99     // primitive is a good starting point.

```



Стандартните Java библиотеки не предлагат функционалност за по-ефективно управление на входната информация. Програмистите сами трябва да поемат грижата за информация, която постъпва в системата. Този подход може да бъде доста объркващ и да създаде много затруднения при работата с входната информация.



```

62     run ant colony optimization.
63     */
64     for (int i = 0; i < Util.NUMBER_OF_REPETITIONS; i++) {
65         graph.resetTau();
66         AntColony4EIA colony = new AntColony4EIA(graph, Util.NUMBER_OF_ANTS,
67             Util.NUMBER_OF_ITERATIONS);
68         colony.start();
69     }
70 }
71
72 /**
73  * Single entry point of the program.
74  *
75  * @param args
76  *         Command line parameters.
77  * @throws Exception
78  *         Exception program stop.
79  */
80 public static void main(String[] args) throws Exception {
81     File input = new File(args[0]);
82     File output = new File(args[1]);
83     String path = output.getCanonicalPath() + "/";
84
85     original = ImageIO.read(input);
86
87     Ellipse.WIDTH = Integer.valueOf(args[4]);
88     Ellipse.HEIGHT = Integer.valueOf(args[5]);
89
90     colors.clear();
91     for (int i = 6; i < args.length; i++) {
92         colors.add(new Color(
93             Integer.parseInt(args[i], 16) | Util.ELLIPSES_ALPHA << 24,
94             true));
95     }
96
97     // TODO Should be some kind of external parameter. The area of the image
98     // divided by the area of the bounding rectangle of the simple graphic
99     // primitive is a good starting point.

```

За преодоляването на всички тези трудности може да се използва библиотеката Apache Commons CLI, която дава множество възможности за организирано управление на входящите параметри и информацията, която се предава чрез тях.







## COMMONS CLI

### Overview

[Download](#)  
[Changes Report](#)  
[Mailing Lists](#)  
[Issue Tracking](#)  
[Source Repository](#)

## USER GUIDE

[Getting started](#)  
[Usage scenarios](#)  
[Option properties](#)  
[Javadoc \(Current 1.5-SNAPSHOT\)](#)  
[Javadoc \(1.4\)](#)

## PROJECT DOCUMENTATION

### Project Information

#### About

[Summary](#)  
[Team](#)  
[Source Code Management](#)  
[Issue Management](#)  
[Mailing Lists](#)  
[Dependency Information](#)  
[Dependencies](#)  
[Dependency Convergence](#)  
[CI Management](#)  
[Distribution Management](#)

### Project Reports

## COMMONS

[Home](#)  
[License](#)  
[Components](#)  
[Sandbox](#)  
[Dormant](#)

## GENERAL INFORMATION

[Security](#)

# Commons CLI

The Apache Commons CLI library provides an API for parsing command line options passed to programs. It's also able to print help messages detailing the options available for a command line tool.

Commons CLI supports different types of options:

- POSIX like options (ie. `tar -zxvf foo.tar.gz`)
- GNU like long options (ie. `du --human-readable --max-depth=1`)
- Java like properties (ie. `java -Djava.awt.headless=true -Djava.net.useSystemProxies=true Foo`)
- Short options with value attached (ie. `gcc -O2 foo.c`)
- long options with single hyphen (ie. `ant -projecthelp`)

A typical help message displayed by Commons CLI looks like this:

```
usage: ls
-A,--almost-all      do not list implied . and ..
-a,--all              do not hide entries starting with .
-B,--ignore-backups   do not list implied entried ending with ~
-b,--escape           print octal escapes for nongraphic characters
    --block-size <SIZE> use SIZE-byte blocks
-c                   with -lt: sort by, and show, ctime (time of last
                    modification of file status information) with
                    -l:show ctime and sort by name otherwise: sort
                    by ctime
-C                   list entries by columns
```

Check out the [introduction](#) page for a detailed presentation.

## CLI 2?

Commons CLI 1.0 was formed from the merger of ideas and code from three different libraries - Werken, Avalon and Optz. In dealing with the bugs and the feature requests a freshly designed and not backwards compatible CLI 2 was created in 2004, but never finished or released.

The current plan is to continue to maintain the 1.x line. The CLI2 work may be found in the Commons Sandbox.

## Documentation

A full [User's Guide](#) is available as are various [project reports](#).

The Javadoc API documents are available online:



Анализът на входната информация започва със създаването на обект съдържащ всички възможни опции. Този обект е наличен от пакета: `org.apache.commons.cli.Options`



```

55         neighbours[i][j] = Math.abs(ellipses.get(i).color.getRGB()
56             - ellipses.get(j).color.getRGB());
57         neighbours[j][i] = neighbours[i][j];
58     }
59 }
60 AntGraph graph = new AntGraph(ellipses.size(), neighbours);
61
62 /*
63  * Run ant colony optimization.
64  */
65 for (int i = 0; i < Util.NUMBER_OF_REPETITIONS; i++) {
66     graph.resetTau();
67     AntColony4EIA colony = new AntColony4EIA(graph, Util.NUMBER_OF_ANTS,
68         Util.NUMBER_OF_ITERATIONS);
69     colony.start();
70 }
71 }
72
73 /**
74  * Single entry point of the program.
75  *
76  * @param args
77  *         Command line parameters.
78  * @throws Exception
79  *         Exception program stop.
80  */
81 public static void main(String[] args) throws Exception {
82     /* Handling command line arguments with a library. */
83     Options options = new Options();
84
85     File input = new File(args[0]);
86     File output = new File(args[1]);
87     String path = output.getCanonicalPath() + "/";
88
89     original = ImageIO.read(input);
90
91     Ellipse.WIDTH = Integer.valueOf(args[4]);

```

При интерфейс от командния ред най-важното е да се добави опция за извикване на помощна информация. Прието е тази опция да бъде `h` или думата `help`. Библиотеката `CLI` позволява кратка форма на опцията, дълга форма на опцията, опцията да има или няма аргументи и конкретен текст обясняващ самата опция.





```

59     }
60 }
61 AntGraph graph = new AntGraph(ellipses.size(), neighbours);
62
63 /*
64  * Run ant colony optimization.
65  */
66 for (int i = 0; i < Util.NUMBER_OF_REPETITIONS; i++) {
67     graph.resetTau();
68     AntColony4EIA colony = new AntColony4EIA(graph, Util.NUMBER_OF_ANTS,
69         Util.NUMBER_OF_ITERATIONS);
70     colony.start();
71 }
72 }
73
74 /**
75  * Single entry point of the program.
76  *
77  * @param args
78  *         Command line parameters.
79  * @throws Exception
80  *         Exception program stop.
81  */
82 public static void main(String[] args) throws Exception {
83     /* Handling command line arguments with a library. */
84     Options options = new Options();
85
86     options.addOption(new Option("h", "help", false, "Help screen.));
87
88     File input = new File(args[0]);
89     File output = new File(args[1]);
90     String path = output.getCanonicalPath() + "/";
91
92     original = ImageIO.read(input);
93
94     Ellipse.WIDTH = Integer.valueOf(args[4]);
95     Ellipse.HEIGHT = Integer.valueOf(args[5]);

```



Тъй като в главната функция попада масив със символни низове, то тези символни низове трябва да бъдат раздробени на части и от тях да се извлече информацията за опциите и евентуално стойностите, които те определят. Това с тава с обект интерпретатор към който се подава обекта с предварително дефинираните опции.



```
*Main.java 68      */
69      for (int i = 0; i < Util.NUMBER_OF_REPETITIONS; i++) {
70          graph.resetTau();
71          AntColony4EIA colony = new AntColony4EIA(graph, Util.NUMBER_OF_ANTS,
72              Util.NUMBER_OF_ITERATIONS);
73          colony.start();
74      }
75  }
76
77  /**
78   * Single entry point of the program.
79   *
80   * @param args
81   *         Command line parameters.
82   * @throws Exception
83   *         Exception program stop.
84   */
85  public static void main(String[] args) throws Exception {
86      /* Handling command line arguments with a library. */
87      Options options = new Options();
88
89      options.addOption(new Option("h", "help", false, "Help screen.));
90
91      /* Parse command line arguments. */
92      CommandLineParser parser = new DefaultParser();
93      CommandLine commands = parser.parse(options, args);
94
95      File input = new File(args[0]);
96      File output = new File(args[1]);
97      String path = output.getCanonicalPath() + "/";
98
99      original = ImageIO.read(input);
100
101      Ellipse.WIDTH = Integer.valueOf(args[4]);
102      Ellipse.HEIGHT = Integer.valueOf(args[5]);
103
104      colors.clear();
```

Всяка възможна опция в командния ред трябва да бъде прихваната и обработена съобразно заложената логика в софтуера. Автоматично генериране на помощното съдържание става с обект от клас HelpFormatter. След визуализацията на помощната информация, изпълнението на програмата приключва.





```

83  *
84  * @param args
85  *         Command line parameters.
86  * @throws Exception
87  *         Exception program stop.
88  */
89  public static void main(String[] args) throws Exception {
90      /* Handling command line arguments with a library. */
91      Options options = new Options();
92
93      options.addOption(new Option("h", "help", false, "Help screen.));
94
95      /* Parse command line arguments. */
96      CommandLineParser parser = new DefaultParser();
97      CommandLine commands = parser.parse(options, args);
98
99      /* If help is requested print it and quit the program. */
100     if (commands.hasOption("help") == true) {
101         printAbout();
102         System.out.println();
103         (new HelpFormatter()).printHelp(
104             "java -jar Ellipses-Image-Approximator-all.jar", options,
105             true);
106         System.out.println();
107         System.exit(0);
108     }
109
110     File input = new File(args[0]);
111     File output = new File(args[1]);
112     String path = output.getCanonicalPath() + "/";
113
114     original = ImageIO.read(input);
115
116     Ellipse.WIDTH = Integer.valueOf(args[4]);
117     Ellipse.HEIGHT = Integer.valueOf(args[5]);
118
119     colors.clear();

```



При софтуер само с текстов интерфейс е важно ясно да се обозначава собственика на авторските права и при какви юридически условия (лиценз) се използва софтуерът.



```

76 }
77
78 /**
79  * Print about information on the standard output.
80  */
81 private static void printAbout() {
82     System.out.println("*****");
83     System.out.println(" Ellipses Image Approximator version 0.0.1");
84     System.out.println(" Copyrights (C) 2016-2020 Velbazhd Software LLC");
85     System.out.println(" developed by Todor Balabanov ( toдор.balabanov@gmail.com )");
86     System.out.println(" Sofia, Bulgaria");
87     System.out.println(" This software is partially supported by the Bulgarian Ministry of Education");
88     System.out.println(" and Science (contract D01-205/23.11.2018) under the National Scientific");
89     System.out.println(" Program \"Information and Communication Technologies for a Single Digital");
90     System.out.println(" Market in Science, Education and Security (ICTinSES)\", approved by");
91     System.out.println(" DCM # 577/17.08.2018.");
92     System.out.println(" This program is free software: you can redistribute it and/or modify");
93     System.out.println(" it under the terms of the GNU General Public License as published by");
94     System.out.println(" the Free Software Foundation, either version 3 of the License, or");
95     System.out.println(" (at your option) any later version.");
96     System.out.println(" This program is distributed in the hope that it will be useful,");
97     System.out.println(" but WITHOUT ANY WARRANTY; without even the implied warranty of");
98     System.out.println(" MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the");
99     System.out.println(" GNU General Public License for more details.");
100    System.out.println(" You should have received a copy of the GNU General Public License");
101    System.out.println(" along with this program. If not, see <http://www.gnu.org/licenses/>.");
102    System.out.println("*****");
103 }
104
105 /**
106  * Single entry point of the program.

```



В резултат на така направените промени, при извикване на програмата с опцията -h се визуализира помощната информация, която включва информация за продукта и начин за използване.



```
MACMINI:libs todorbalabanov$ java -jar Ellipses-Image-Approximator-all.jar -help
*****
* Ellipses Image Approximator version 0.0.1 *
* Copyrights (C) 2016-2020 Velbazhd Software LLC *
* *
* developed by Todor Balabanov ( todor.balabanov@gmail.com ) *
* Sofia, Bulgaria *
* *
* This software is partially supported by the Bulgarian Ministry of Education *
* and Science (contract D01-205/23.11.2018) under the National Scientific *
* Program "Information and Communication Technologies for a Single Digital *
* Market in Science, Education and Security (ICTinSES)", approved by *
* DCM # 577/17.08.2018. *
* *
* This program is free software: you can redistribute it and/or modify *
* it under the terms of the GNU General Public License as published by *
* the Free Software Foundation, either version 3 of the License, or *
* (at your option) any later version. *
* *
* This program is distributed in the hope that it will be useful, *
* but WITHOUT ANY WARRANTY; without even the implied warranty of *
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
* GNU General Public License for more details. *
* *
* You should have received a copy of the GNU General Public License *
* along with this program. If not, see <http://www.gnu.org/licenses/>. *
* *
*****

usage: java -jar Ellipses-Image-Approximator-all.jar [-h]
-h,--help Help screen.

MACMINI:libs todorbalabanov$
```



Най-важният аргумент, без който програмата не може да извърши изчисленията си, е пътят и името на файла с изображението. Този параметър е задължителен и ако липсва изпълнението на програмата трябва да се прекрати.



```

112  * Single entry point of the program.
113  *
114  * @param args
115  *         Command line parameters.
116  * @throws Exception
117  *         Exception program stop.
118  */
119  public static void main(String[] args) throws Exception {
120      /* Handling command line arguments with a library. */
121      Options options = new Options();
122
123      options.addOption(new Option("h", "help", false, "Help screen.));
124
125      options.addOption(Option.builder("input").argName("file").hasArg()
126          .valueSeparator().desc("Image path and file name.").build());
127
128      /* Parse command line arguments. */
129      CommandLineParser parser = new DefaultParser();
130      CommandLine commands = parser.parse(options, args);
131
132      /* If help is requested print it and quit the program. */
133      if (commands.hasOption("help") == true) {
134          printAbout();
135          System.out.println();
136          (new HelpFormatter()).printHelp(
137              "java -jar Ellipses-Image-Approximator-all.jar", options,
138              true);
139          System.out.println();
140          System.exit(0);
141      }
142
143      File input = new File(args[0]);
144      File output = new File(args[1]);
145      String path = output.getCanonicalPath() + "/";
146
147      original = ImageIO.read(input);
148

```

Файловите операции се смятат ненадеждни и е възможно възникване на изключителни ситуации. В случая с проблемен входен файл няма вариант програмата да извърши нужните пресмятания и възникналото изключение ще доведе до моментално прекратяване на изпълнението.





```

150 options.addOption(new Option("h", "help", false, "Help screen.));
151
152 options.addOption(Option.builder("input").argName("file").hasArg()
153     .valueSeparator().desc("Image path and file name.").build());
154
155 /* Parse command line arguments. */
156 CommandLineParser parser = new DefaultParser();
157 CommandLine commands = parser.parse(options, args);
158
159 /* If help is requested print it and quit the program. */
160 if (commands.hasOption("help") == true) {
161     printAbout();
162     System.out.println();
163     (new HelpFormatter()).printHelp(
164         "java -jar Ellipses-Image-Approximator-all.jar", options,
165         true);
166     System.out.println();
167     System.exit(0);
168 }
169
170 /* Associate input file. */
171 File input = null;
172 if (commands.hasOption("input") == true) {
173     input = new File(commands.getOptionValue("input"));
174 } else {
175     System.out.println("Input file name is missing!");
176     System.out.println();
177     (new HelpFormatter()).printHelp(
178         "java -jar Ellipses-Image-Approximator-all.jar", options,
179         true);
180     System.out.println();
181     System.exit(0);
182 }
183
184 File output = new File(args[1]);
185 String path = output.getCanonicalPath() + "/";
186

```

Директорията за резултатните файлове е пожелателен аргумент и когато не е определена се използва текущо активната директория.



```

140  *
141  * @param args
142  *      Command line parameters.
143  * @throws Exception
144  *      Exception program stop.
145  */
146  public static void main(String[] args) throws Exception {
147      /* Handling command line arguments with a library. */
148      Options options = new Options();
149
150      options.addOption(new Option("h", "help", false, "Help screen.));
151
152      options.addOption(Option.builder("input").argName("file").hasArg()
153          .valueSeparator().desc("Image path and file name.").build());
154
155      options.addOption(Option.builder("output").argName("folder").hasArg()
156          .valueSeparator()
157          .desc("Output folder path (current folder is a default value).")
158          .build());
159
160      /* Parse command line arguments. */
161      CommandLineParser parser = new DefaultParser();
162      CommandLine commands = parser.parse(options, args);
163
164      /* If help is requested print it and quit the program. */
165      if (commands.hasOption("help") == true) {
166          printAbout();
167          System.out.println();
168          (new HelpFormatter()).printHelp(
169              "java -jar Ellipses-Image-Approximator-all.jar", options,
170              true);
171          System.out.println();
172          System.exit(0);
173      }
174
175      /* Associate input file. */
176      File input = null;

```



Когато програмата генерира твърде много файлове с резултати е добре да съществува възможност за записването на тези файлове в конкретна директория, предназначена за тази цел. Ако такава не е указана, то файловете ще се появят в директорията, която е активна за изпълнявания процес в операционната система.



```

173     }
174
175     /* Associate input file. */
176     File input = null;
177     if (commands.hasOption("input") == true) {
178         input = new File(commands.getOptionValue("input"));
179     } else {
180         System.out.println("Input file name is missing!");
181         System.out.println();
182         (new HelpFormatter()).printHelp(
183             "java -jar Ellipses-Image-Approximator-all.jar", options,
184             true);
185         System.out.println();
186         System.exit(0);
187     }
188
189     /* Associate output folder. */
190     File output = null;
191     if (commands.hasOption("output") == true) {
192         output = new File(commands.getOptionValue("output"));
193     } else {
194         output = new File("./");
195     }
196
197     String path = output.getCanonicalPath() + "/";
198
199     original = ImageIO.read(input);
200
201     Ellipse.WIDTH = Integer.valueOf(args[4]);
202     Ellipse.HEIGHT = Integer.valueOf(args[5]);
203
204     colors.clear();
205     for (int i = 6; i < args.length; i++) {
206         colors.add(new Color(
207             Integer.parseInt(args[i], 16) | Util.ELLIPSES_ALPHA << 24,
208             true));
209     }

```

Ширината и височината на използваните елипси е също параметър, който се подава при извикването. Стойността по подразбиране е единица, което би съответствало на един пиксел в растерното изображение.





```

145  */
146  public static void main(String[] args) throws Exception {
147      /* Handling command line arguments with a library. */
148      Options options = new Options();
149
150      options.addOption(new Option("help", false, "Help screen.));
151
152      options.addOption(Option.builder("input").argName("file").hasArg()
153          .valueSeparator().desc("Image path and file name.").build());
154
155      options.addOption(Option.builder("output").argName("folder").hasArg()
156          .valueSeparator()
157          .desc("Output folder path (default value current folder).")
158          .build());
159
160      options.addOption(Option.builder("ellipse_width").argName("number").hasArg()
161          .valueSeparator().desc("Ellipse width (default value 1).").build());
162
163      options.addOption(Option.builder("ellipse_height").argName("number").hasArg()
164          .valueSeparator().desc("Ellipse width (default value 1).").build());
165
166      /* Parse command line arguments. */
167      CommandLineParser parser = new DefaultParser();
168      CommandLine commands = parser.parse(options, args);
169
170      /* If help is requested print it and quit the program. */
171      if (commands.hasOption("help") == true) {
172          printAbout();
173          System.out.println();
174          (new HelpFormatter()).printHelp(
175              "java -jar Ellipses-Image-Approximator-all.jar", options,
176              true);
177          System.out.println();
178          System.exit(0);
179      }
180
181      /* Associate input file. */

```

При трансформацията на текст към число е възможно да възникнат изключителни ситуации, но тъй като параметрите за елипсата са от критично значение, то аварийното прекъсване на програмата е приемлив сценарий.



Списъкът с цветове може да варира и подаването му става с шестнадесетични числа, които са разделени със запетая. Това представяне може да създаде проблеми при прочитането, но без списък от цветове програмата не би могла да извърши нужните изчисления.





```

152 options.addOption(Option.builder("input").argName("file").hasArg()
153     .valueSeparator().desc("Image path and file name.").build());
154
155 options.addOption(Option.builder("output").argName("folder").hasArg()
156     .valueSeparator()
157     .desc("Output folder path (default value current folder).")
158     .build());
159
160 options.addOption(Option.builder("ellipse_width").argName("number")
161     .hasArg().valueSeparator()
162     .desc("Ellipse width (default value 1).").build());
163
164 options.addOption(Option.builder("ellipse_height").argName("number")
165     .hasArg().valueSeparator()
166     .desc("Ellipse height (default value 1).").build());
167
168 options.addOption(Option.builder("colors").argName("hex1,hex2,hex3,...")
169     .hasArg().valueSeparator()
170     .desc("Set of colors as comma separeated list of RGB hexadecimal numbers.")
171     .build());
172
173 /* Parse command line arguments. */
174 CommandLineParser parser = new DefaultParser();
175 CommandLine commands = parser.parse(options, args);
176
177 /* If help is requested print it and quit the program. */
178 if (commands.hasOption("help") == true) {
179     printAbout();
180     System.out.println();
181     (new HelpFormatter()).printHelp(
182         "java -jar Ellipses-Image-Approximator-all.jar", options,
183         true);
184     System.out.println();
185     System.exit(0);
186 }
187
188 /* Associate input file. */

```

Стойностите за цветовете се раздробяват по разделителната запетая и се трансформират до обекти тип Color с добавяне на стойност за прозрачност.



```

212     if (commands.hasOption("ellipse_width") == true) {
213         Ellipse.WIDTH = Integer
214             .valueOf(commands.getOptionValue("ellipse_width"));
215     } else {
216         Ellipse.WIDTH = 1;
217     }
218
219     /* Set ellipse height. */
220     if (commands.hasOption("ellipse_height") == true) {
221         Ellipse.HEIGHT = Integer
222             .valueOf(commands.getOptionValue("ellipse_height"));
223     } else {
224         Ellipse.HEIGHT = 1;
225     }
226
227     /* Parse hexadecimal values for the colors.. */
228     colors.clear();
229     if (commands.hasOption("colors") == true) {
230         String[] values = commands.getOptionValue("colors").split(",");
231         for (String value : values) {
232             colors.add(new Color(
233                 Integer.parseInt(value, 16) | Util.ELLIPSES_ALPHA << 24,
234                 true));
235         }
236     }
237
238     /* Read input image. */
239     original = ImageIO.read(input);
240
241     // TODO Should be some kind of external parameter. The area of the image
242     // divided by the area of the bounding rectangle of the simple graphic
243     // primitive is a good starting point.
244     EllipseListChromosome
245         .AVERAGE_LENGTH((original.getWidth() * original.getHeight())
246             / (Ellipse.WIDTH * Ellipse.HEIGHT));
247
248     Population initial = Util.randomInitialPopulation(original, colors,

```



По настояще софтуерът предлага две възможности за оптимизация – чрез генетични алгоритми и чрез колония от мравки. За тези две възможности е съществено да се добавят опции с които да бъдат включвани. Подразбиращото се състояние е изключени.



```

173      /* Parse command line arguments. */
174      CommandLineParser parser = new DefaultParser();
175      CommandLine commands = parser.parse(options, args);
176
177      /* If help is requested print it and quit the program. */
178      if (commands.hasOption("help") == true) {
179          printAbout();
180          System.out.println();
181          (new HelpFormatter()).printHelp(
182              "java -jar Ellipses-Image-Approximator-all.jar", options,
183              true);
184          System.out.println();
185          System.exit(0);
186      }
187
188      /* Switch on genetic algorithm optimization. */
189      boolean useGeneticAlgorithmOptimization = false;
190      if (commands.hasOption("ga") == true) {
191          useGeneticAlgorithmOptimization = true;
192      }
193
194      /* Switch on ant colony optimization. */
195      boolean useAntColonyOptimization = false;
196      if (commands.hasOption("aco") == true) {
197          useAntColonyOptimization = true;
198      }
199
200      /* Associate input file. */
201      File input = null;
202      if (commands.hasOption("input") == true) {
203          input = new File(commands.getOptionValue("input"));
204      } else {
205          System.out.println("Input file name is missing!");
206          System.out.println();
207          (new HelpFormatter()).printHelp(
208              "java -jar Ellipses-Image-Approximator-all.jar", options,
209              true);

```

Двете опции нямат аргументи и отсъствието им от командния ред означава, че са в състояние изключени.





```

146= public static void main(String[] args) throws Exception {
147     /* Handling command line arguments with a library. */
148     Options options = new Options();
149
150     options.addOption(new Option("help", false, "Help screen."));
151
152     options.addOption(new Option("ga", false,
153         "Switch on genetic algorithm optimization (default value off)."));
154
155     options.addOption(new Option("aco", false,
156         "Switch on ant colony optimization (default value off)."));
157
158     options.addOption(Option.builder("input").argName("file").hasArg()
159         .valueSeparator().desc("Image path and file name.").build());
160
161     options.addOption(Option.builder("output").argName("folder").hasArg()
162         .valueSeparator()
163         .desc("Output folder path (default value current folder).")
164         .build());
165
166     options.addOption(Option.builder("ellipse_width").argName("number")
167         .hasArg().valueSeparator()
168         .desc("Ellipse width (default value 1).").build());
169
170     options.addOption(Option.builder("ellipse_height").argName("number")
171         .hasArg().valueSeparator()
172         .desc("Ellipse height (default value 1).").build());
173
174     options.addOption(Option.builder("colors").argName("hex1,hex2,hex3,...")
175         .hasArg().valueSeparator()
176         .desc("Set of colors as comma separeated list of RGB hexadecimal numbers.")
177         .build());
178
179     /* Parse command line arguments. */
180     CommandLineParser parser = new DefaultParser();
181     CommandLine commands = parser.parse(options, args);
182

```

Размерът на популацията в генетичния алгоритъм и времето за оптимизация са два други параметъра, които би било от голяма полза да се подават през командния ред.





```

240         .valueOf(commands.getOptionValue("ellipse_height"));
241     }
242
243     // TODO Ellipse alpha value should be input argument.
244
245     /* Parse hexadecimal values for the colors.. */
246     colors.clear();
247     if (commands.hasOption("colors") == true) {
248         String[] values = commands.getOptionValue("colors").split(",");
249         for (String value : values) {
250             colors.add(new Color(
251                 Integer.parseInt(value, 16) | Util.ELLIPSES_ALPHA << 24,
252                 true));
253         }
254     }
255
256     /* Set population size for the genetic algorithm. */
257     int gaPopulationSize = 0;
258     if (commands.hasOption("ga_population_size") == true) {
259         gaPopulationSize = Integer
260             .valueOf(commands.getOptionValue("ga_population_size"));
261     }
262
263     /* Set genetic algorithm optimization time. */
264     int gaOptimizationTime = 0;
265     if (commands.hasOption("ga_optimization_time") == true) {
266         gaOptimizationTime = Integer
267             .valueOf(commands.getOptionValue("ga_optimization_time"));
268     }
269
270     /* Read input image. */
271     original = ImageIO.read(input);
272
273     // TODO Should be some kind of external parameter. The area of the image
274     // divided by the area of the bounding rectangle of the simple graphic
275     // primitive is a good starting point.
276     EllipseListChromosome

```



Трудно е да се подбере някаква разумна стойност, която да е по подразбиране за размер на популацията и времето за оптимизация. Поради тази причина подразбиращите се стойности са нулеви, което би довело до мигновено приключване на програмата.



```

163         .desc("Output folder path (default value current folder).")
164         .build());
165
166     options.addOption(Option.builder("ellipse_width").argName("number")
167         .hasArg().valueSeparator()
168         .desc("Ellipse width (default value 1).").build());
169
170     options.addOption(Option.builder("ellipse_height").argName("number")
171         .hasArg().valueSeparator()
172         .desc("Ellipse height (default value 1).").build());
173
174     options.addOption(Option.builder("colors").argName("hex1,hex2,hex3,...")
175         .hasArg().valueSeparator()
176         .desc("Set of colors as comma separeated list of RGB hexadecimal numbers.")
177         .build());
178
179     options.addOption(Option.builder("ga_population_size").argName("number")
180         .hasArg().valueSeparator()
181         .desc("Genetic algorithm population size (default value 0).").build());
182
183     options.addOption(Option.builder("ga_optimization_time").argName("number")
184         .hasArg().valueSeparator()
185         .desc("Genetic algorithm optimization time in seconds (default value 0).").build());
186
187     /* Parse command line arguments. */
188     CommandLineParser parser = new DefaultParser();
189     CommandLine commands = parser.parse(options, args);
190
191     /* If help is requested print it and quit the program. */
192     if (commands.hasOption("help") == true) {
193         printAbout();
194         System.out.println();
195         (new HelpFormatter()).printHelp(
196             "java -jar Ellipses-Image-Approximator-all.jar", options,
197             true);
198         System.out.println();
199         System.exit(0);

```

Тъй като отделните графични примитиви ще се припокриват е разумно да се добави някаква степен на прозрачност, така че да се онагледят факта за смесването на боите от различни цветове. Това би дало по-реалистична представа за фактическото изрисуване на картината.





```
237 // Set ellipse width. */
238 Ellipse.WIDTH = 1;
239 if (commands.hasOption("ellipse_width") == true) {
240     Ellipse.WIDTH = Integer
241         .valueOf(commands.getOptionValue("ellipse_width"));
242 }
243
244 /* Set ellipse height. */
245 Ellipse.HEIGHT = 1;
246 if (commands.hasOption("ellipse_height") == true) {
247     Ellipse.HEIGHT = Integer
248         .valueOf(commands.getOptionValue("ellipse_height"));
249 }
250
251 /* Ellipse alpha level. */
252 Ellipse.ALPHA = 0xFF;
253 if (commands.hasOption("ellipse_alpha") == true) {
254     Ellipse.ALPHA = Integer
255         .valueOf(commands.getOptionValue("ellipse_alpha"));
256 }
257
258 /* Parse hexadecimal values for the colors.. */
259 colors.clear();
260 if (commands.hasOption("colors") == true) {
261     String[] values = commands.getOptionValue("colors").split(",");
262     for (String value : values) {
263         colors.add(new Color(
264             Integer.parseInt(value, 16) | Util.ELLIPSES_ALPHA << 24,
265             true));
266     }
267 }
268
269 /* Set population size for the genetic algorithm. */
270 int gaPopulationSize = 0;
271 if (commands.hasOption("ga_population_size") == true) {
272     gaPopulationSize = Integer
273         .valueOf(commands.getOptionValue("ga_population_size"));
274 }
```

Нивата в алфа канала на ARGB цветовата схема са от 0 до 255, тъй като се използват 8 бита. По подразбиране е най-високото ниво, където информацията в алфа канала се игнорира.





```

103
166 options.addOption(Option.builder("ellipse_width").argName("number")
167     .hasArg().valueSeparator()
168     .desc("Ellipse width (default value 1).").build());
169
170 options.addOption(Option.builder("ellipse_height").argName("number")
171     .hasArg().valueSeparator()
172     .desc("Ellipse height (default value 1).").build());
173
174 options.addOption(Option.builder("ellipse_alpha").argName("number")
175     .hasArg().valueSeparator()
176     .desc("Ellipse alpha channel value between 0 and 255 (default value 255).")
177     .build());
178
179 options.addOption(Option.builder("colors").argName("hex1,hex2,hex3,...")
180     .hasArg().valueSeparator()
181     .desc("Set of colors as comma separeated list of RGB hexadecimal numbers.")
182     .build());
183
184 options.addOption(Option.builder("ga_population_size").argName("number")
185     .hasArg().valueSeparator()
186     .desc("Genetic algorithm population size (default value 0).")
187     .build());
188
189 options.addOption(Option.builder("ga_optimization_time")
190     .argName("number").hasArg().valueSeparator()
191     .desc("Genetic algorithm optimization time in seconds (default value 0).")
192     .build());
193
194 /* Parse command line arguments. */
195 CommandLineParser parser = new DefaultParser();
196 CommandLine commands = parser.parse(options, args);
197
198 /* If help is requested print it and quit the program. */
199 if (commands.hasOption("help") == true) {
200     printAbout();
201     System.out.println();
202     new HelpFormatter().printHelpOf

```



Генетичният алгоритъм има серия параметри, които е добре да бъдат подадени като стойности от командния ред. В тази група са нивата за кръстосване, мутация, четността на родителите и нивото на елит.



```

278 if (commands.hasOption("ga_population_size") == true) {
279     gaPopulationSize = Integer
280         .valueOf(commands.getOptionValue("ga_population_size"));
281 }
282
283 /* Set genetic algorithm tournament selection arity. */
284 int gaTournamentArity = 2;
285 if (commands.hasOption("ga_tournament_arity") == true) {
286     gaTournamentArity = Integer
287         .valueOf(commands.getOptionValue("ga_tournament_arity"));
288 }
289
290 /* Set genetic algorithm crossover rate. */
291 double gaCrossoverRate = 0.9;
292 if (commands.hasOption("ga_crossover_rate") == true) {
293     gaCrossoverRate = Double
294         .valueOf(commands.getOptionValue("ga_crossover_rate"));
295 }
296
297 /* Set genetic algorithm mutation rate. */
298 double gaMutationRate = 0.01;
299 if (commands.hasOption("ga_mutation_rate") == true) {
300     gaMutationRate = Double
301         .valueOf(commands.getOptionValue("ga_mutation_rate"));
302 }
303
304 /* Set genetic algorithm elitism rate. */
305 double gaElitismRate = 1.0 / gaPopulationSize;
306 if (commands.hasOption("ga_elitism_rate") == true) {
307     gaElitismRate = Double
308         .valueOf(commands.getOptionValue("ga_elitism_rate"));
309 }
310
311 /* Set genetic algorithm optimization time. */
312 int gaOptimizationTime = 0;
313 if (commands.hasOption("ga_optimization_time") == true) {
314     gaOptimizationTime = Integer

```

Подразбиращите се стойности за кръстосване и мутация са съобразени с препоръките от емпиричните изследвания. Трудно е да се определи подразбираща се стойност за елита и поради тази причина е разумно да се вземе поне един индивид от популацията.





```

182         .build());
183
184     options.addOption(Option.builder("ga_population_size").argName("number")
185         .hasArg().valueSeparator()
186         .desc("Genetic algorithm population size (default value 0).")
187         .build());
188
189     options.addOption(Option.builder("ga_tournament_arity")
190         .argName("number").hasArg().valueSeparator()
191         .desc("Genetic algorithm tournament selection arity (default value 2).")
192         .build());
193
194     options.addOption(Option.builder("ga_crossover_rate").argName("number")
195         .hasArg().valueSeparator()
196         .desc("Genetic algorithm crossover rate from 0.0. to 1.0 (default value 0.9).")
197         .build());
198
199     options.addOption(Option.builder("ga_mutation_rate").argName("number")
200         .hasArg().valueSeparator()
201         .desc("Genetic algorithm mutation rate from 0.0. to 1.0 (default value 0.01).")
202         .build());
203
204     options.addOption(Option.builder("ga_elitism_rate").argName("number")
205         .hasArg().valueSeparator()
206         .desc("Genetic algorithm elitism rate from 0.0. to 1.0 (default value one individual).")
207         .build());
208
209     options.addOption(Option.builder("ga_optimization_time")
210         .argName("number").hasArg().valueSeparator()
211         .desc("Genetic algorithm optimization time in seconds (default value 0).")
212         .build());
213
214     /* Parse command line arguments. */
215     CommandLineParser parser = new DefaultParser();
216     CommandLine commands = parser.parse(options, args);
217
218     /* If help is requested print it and quit the program. */

```

При хромозоми представляващи инструкции е често срещана практика дължината да варира. Поради тази причина е удачно средният размер на хромозомите да се определя с параметър. При лисата на стойност е разумно да се използва същата по площ елипси, каквато е площта на цялото изображение.





```

291     String[] values = commands.getOptionValue("colors").split(",");
292     for (String value : values) {
293         colors.add(new Color(
294             Integer.parseInt(value, 16) | Ellipse.ALPHA << 24,
295             true));
296     }
297 }
298
299 /* Set population size for the genetic algorithm. */
300 int gaPopulationSize = 0;
301 if (commands.hasOption("ga_population_size") == true) {
302     gaPopulationSize = Integer
303         .valueOf(commands.getOptionValue("ga_population_size"));
304 }
305
306 /* Set chromosome average size for the genetic algorithm. */
307 int gaChromosomeAverageSize = (original.getWidth()
308     * original.getHeight()) / (Ellipse.WIDTH * Ellipse.HEIGHT);
309 if (commands.hasOption("ga_chromosome_size") == true) {
310     gaPopulationSize = Integer
311         .valueOf(commands.getOptionValue("ga_chromosome_size"));
312 }
313
314 /* Set genetic algorithm tournament selection arity. */
315 int gaTournamentArity = 2;
316 if (commands.hasOption("ga_tournament_arity") == true) {
317     gaTournamentArity = Integer
318         .valueOf(commands.getOptionValue("ga_tournament_arity"));
319 }
320
321 /* Set genetic algorithm crossover rate. */
322 double gaCrossoverRate = 0.9;
323 if (commands.hasOption("ga_crossover_rate") == true) {
324     gaCrossoverRate = Double
325         .valueOf(commands.getOptionValue("ga_crossover_rate"));
326 }
327

```



Трудно е да се даде оценка за размера на подразбиращата се стойност, но може да се укаже, че тя зависи от размера на обработваното изображение и от размерите на графичния примитив.



```
178
179 options.addOption(Option.builder("colors").argName("hex1,hex2,hex3,...")
180     .hasArg().valueSeparator()
181     .desc("Set of colors as comma separeated list of RGB hexadecimal numbers.")
182     .build());
183
184 options.addOption(Option.builder("ga_population_size").argName("number")
185     .hasArg().valueSeparator()
186     .desc("Genetic algorithm population size (default value 0).")
187     .build());
188
189 options.addOption(Option.builder("ga_chromosome_size").argName("number")
190     .hasArg().valueSeparator()
191     .desc("Genetic algorithm chromosome average size (default value depends of the image siz
192     .build());
193
194 options.addOption(Option.builder("ga_tournament_arity")
195     .argName("number").hasArg().valueSeparator()
196     .desc("Genetic algorithm tournament selection arity (default value 2).")
197     .build());
198
199 options.addOption(Option.builder("ga_crossover_rate").argName("number")
200     .hasArg().valueSeparator()
201     .desc("Genetic algorithm crossover rate from 0.0. to 1.0 (default value 0.9).")
202     .build());
203
204 options.addOption(Option.builder("ga_mutation_rate").argName("number")
205     .hasArg().valueSeparator()
206     .desc("Genetic algorithm mutation rate from 0.0. to 1.0 (default value 0.01).")
207     .build());
208
209 options.addOption(Option.builder("ga_elitism_rate").argName("number")
210     .hasArg().valueSeparator()
211     .desc("Genetic algorithm elitism rate from 0.0. to 1.0 (default value one individual).")
212     .build());
213
214 options.addOption(Option.builder("ga_optimization_time")
```

След като софтуерът вече се извиква по друг стандарт, то е важно да се осъвремени и публичната информация за използване на програмата.





```
1 # Ellipses Image Approximator
2
3 It is an attempt to approximate full colors image with limited set of colors for simple ellipses.
4
5 ## Build Instructions
6
7 ./gradlew jarAll
8
9 ## Run Examples
10
11 java -jar Ellipses-Image-Approximator-all.jar -input ../../input/0001.jpg -output ~/Desktop/ -
  ellipse_width 3 -ellipse_height 19 -ellipse_alpha 216 -colors 000000,808080,C0C0C0,FFFFFF,
  800000,FF0000,808000,FFFF00,008000,00FF00,008080,00FFFF,000080,0000FF,800080,FF00FF -ga -
  ga_population_size 31 -ga_chromosome_size 100 -ga_tournament_arity 2 -ga_crossover_rate 0.9 -
  ga_mutation_rate 0.1 -ga_elitism_rate 0.1 -ga_optimization_time 100
12
13 java -jar Ellipses-Image-Approximator-all.jar -input ../../input/0009.jpg -output ~/Desktop/ -
  ellipse_width 5 -ellipse_height 19 -ellipse_alpha 216 -colors 000000,808080,C0C0C0,FFFFFF,
  800000,FF0000,808000,FFFF00,008000,00FF00,008080,00FFFF,000080,0000FF,800080,FF00FF -ga -
  ga_population_size 11 -ga_chromosome_size 100 -ga_tournament_arity 2 -ga_crossover_rate 0.9 -
  ga_mutation_rate 0.1 -ga_elitism_rate 0.1 -ga_optimization_time 7
14
15 ## Acknowledgements
16
17 This software is funded by Velbazhd Software LLC and it is partially supported by the Bulgarian Ministry
  of
18 Education and Science (contract D01-205/23.11.2018) under the National Scientific Program "Information
  and
19 Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)",
20 approved by DCM # 577/17.08.2018.
21
22 [//]: # (This work was supported by a grant of the Bulgarian National Scientific Fund under the grants
  DFNI 02/20 Efficient Parallel Algorithms for Large Scale Computational Problems and DFNI 02/5
  InterCriteria Analysis A New Approach to Decision Making.)
23
24 ![alt tag](http://s4.postimg.org/v4ylmm46l/output_hy_WXCL.gif)
25
```

След добавянето на всички планирани опции, командата за извикване на програмата придобива следния вид.





```

* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*
*****

```

```

usage: java -jar Ellipses-Image-Approximator-all.jar [-aco] [-colors
<hex1,hex2,hex3,...>] [-ellipse_alpha <number>] [-ellipse_height
<number>] [-ellipse_width <number>] [-ga] [-ga_chromosome_size
<number>] [-ga_crossover_rate <number>] [-ga_elitism_rate <number>]
[-ga_mutation_rate <number>] [-ga_optimization_time <number>]
[-ga_population_size <number>] [-ga_tournament_arity <number>]
[-help] [-input <file>] [-output <folder>]

```

```

-aco                Switch on ant colony optimization
                    (default value off).
-colors <hex1,hex2,hex3,...>  Set of colors as comma separeated list
                    of RGB hexadecimal numbers.
-ellipse_alpha <number>      Ellipse alpha channel value between 0
                    and 255 (default value 255).
-ellipse_height <number>     Ellipse height (default value 1).
-ellipse_width <number>      Ellipse width (default value 1).
-ga                    Switch on genetic algorithm optimization
                    (default value off).
-ga_chromosome_size <number>  Genetic algorithm chromosome average
                    size (default value depends of the image
                    size and ellipse size).
-ga_crossover_rate <number>   Genetic algorithm crossover rate from
                    0.0. to 1.0 (default value 0.9).
-ga_elitism_rate <number>     Genetic algorithm elitism rate from 0.0.
                    to 1.0 (default value one individual).
-ga_mutation_rate <number>    Genetic algorithm mutation rate from
                    0.0. to 1.0 (default value 0.01).
-ga_optimization_time <number> Genetic algorithm optimization time in
                    seconds (default value 0).
-ga_population_size <number>  Genetic algorithm population size
                    (default value 0).
-ga_tournament_arity <number> Genetic algorithm tournament selection
                    arity (default value 2).
-help                    Help screen.
-input <file>            Image path and file name.
-output <folder>         Output folder path (default value
                    current folder).

```



След проверка дали промените по кода водят до коректно компилиране и изграждане на изпълнимия файл, може да се пристъпи към публикуване на кода в отдалеченото хранилище.



```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ ./gradlew jarAll
```

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.

Use '--warning-mode all' to show the individual deprecation warnings.

See [https://docs.gradle.org/6.2.1/userguide/command\\_line\\_interface.html#sec:command\\_line\\_warnings](https://docs.gradle.org/6.2.1/userguide/command_line_interface.html#sec:command_line_warnings)

**BUILD SUCCESSFUL** in 3s

2 actionable tasks: 2 up-to-date

```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ █
```

Публикуването в глобалното хранилище преминава през добавяне на файловете за синхронизация, публикуване в локалното хранилище и изпращане до облачната услуга GitHub.





```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ git status
```

```
On branch master
```

```
Your branch is up to date with 'origin/master'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified: README.md
```

```
modified: src/main/java/eu/veldsoft/ellipses/image/approximator/Ellipse.java
```

```
modified: src/main/java/eu/veldsoft/ellipses/image/approximator/Main.java
```

```
modified: src/main/java/eu/veldsoft/ellipses/image/approximator/Util.java
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ git add .
```

```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ git status
```

```
On branch master
```

```
Your branch is up to date with 'origin/master'.
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
modified: README.md
```

```
modified: src/main/java/eu/veldsoft/ellipses/image/approximator/Ellipse.java
```

```
modified: src/main/java/eu/veldsoft/ellipses/image/approximator/Main.java
```

```
modified: src/main/java/eu/veldsoft/ellipses/image/approximator/Util.java
```

```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ git commit -m "CLI was used for the arguments."
```

```
[master eb49d87] CLI was used for the arguments.
```

```
4 files changed, 134 insertions(+), 40 deletions(-)
```

```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ git push
```

```
Counting objects: 14, done.
```

```
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (9/9), done.
```

```
Writing objects: 100% (14/14), 2.27 KiB | 776.00 KiB/s, done.
```

```
Total 14 (delta 6), reused 0 (delta 0)
```

```
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
```

```
To https://github.com/TodorBalabanov/Ellipses-Image-Approximator.git
```

```
268f1ff..eb49d87 master -> master
```

```
MACMINI:Ellipses-Image-Approximator todorbalabanov$ █
```

В представения проект с отворен код има още множество променливи, които трябва да бъдат параметризирани през командния ред, но към настоящия етап направената параметризация е напълно достатъчна.



