

## **МЕТОДИЧКА**

212 – Чернышова – Светлана

### **КОМАНДЫ**

#### ***GSS hight mu\_x mu\_y sigma\_x sigma\_y***

- } Создает Гаусс заданных параметров
- } Где:
  - ❖ hight – целое число, высота Гаусса
  - ❖ mu\_x – целое число, координата вершины Гаусса по x
  - ❖ mu\_y – целое число, координата вершины Гаусса по y
  - ❖ sigma\_x – целое число, растяжение по x
  - ❖ sigma\_y – целое число, растяжение по y
- } Пример: GSS 200 40 50 2 7

#### ***INIT x\_size y\_size***

- } Создает поле заданного размера
- } Где:
  - ❖ x\_size – целое число, размер поля по x
  - ❖ y\_size – целое число, размер поля по y
- } Пример: INIT 100 100
- } Ограничения:
  - ❖ Из соображений здравого смысла размеры поля целые и неотрицательные
  - ❖ Нельзя использовать дважды. У вас ровно одно поле!

#### ***GEN***

- } Добавляет Гауссы на поле
- } Пример: GEN
- } Ограничения:
  - ❖ Команда не может быть использована до хотя бы одной команды GSS, потому что в противном случае ей нечего делать. В этом случае она будет пропущена.
  - ❖ Команда не может быть использована до создания поля: INIT, потому что в противном случае ей некуда будет помещать Гауссы
  - ❖ Нельзя использовать дважды

#### ***CRBMP bmp\_created\_filename.bmp***

- } Создает картинку формата .bmp текущего поля
- } Где:
  - ❖ bmp\_created\_filename.bmp – имя файла, в который пользователь хочет записать картинку
- } Пример: CRBMP bmp\_created\_test11.bmp
- } Ограничения:
  - ❖ Команда не может быть использована до создания поля: INIT, потому что в противном случае ей нечего будет изображать
  - ❖ Команда не может быть использована до команды генерации поля (добавления на него Гауссов) либо команды чтения .bmp, потому что в противном случае поле тривиально и рисовать нечего

## ***RDBMP bmp\_read\_filename.bmp***

- } Читает картинку формата .bmp в текущее поле
- } Где:
  - ❖ bmp\_read\_filename.bmp – имя файла, из которого пользователь хочет прочитать картинку
- } Пример: RDBMP kotik.bmp
- } Ограничения:
  - ❖ Команда не может быть использована после создания поля: INIT, потому что она сама создает поле, соответствующее размеру картинки

## ***GNU gnuplot\_data\_filename.txt Gnuplot\_commands\_filename.txt***

- } Создает два файла формата .txt.  
Один – с непосредственными командами для гнуплота, второй – с данными (содержание поля), по которым гнуплот будет строить картинку.  
Сама картинка строится так:
  - ❖ В командной строке пишутся команды
  - `gnuplot`
  - ❖ После чего запускается гнуплот
  - ❖ Далее пишем
  - `load "Gnuplot_commands_filename.txt"`
  - ❖ После чего создается картинка
  - ❖ Для выхода из гнуплота пишем
  - `exit`
- } Где:
  - ❖ gnuplot\_data\_filename.txt – имя файла для записи данных и построения картинки
  - ❖ Gnuplot\_commands\_filename.txt – имя файла для записи непосредственно команд гнуплота
- } Пример: GNU gnuplot\_data\_test11.txt Gnuplot\_test11.txt
- } Ограничения:
  - ❖ Команда не может быть использована до команды генерации поля (добавления на него Гауссов) либо команды чтения .bmp, потому что в противном случае поле тривиально и рисовать нечего

## ***BIN slice\_hight***

- } Обрезает поле по указанному значению
- } Где:
  - ❖ slice\_hight – целое число, высота среза
- } Пример: BIN 130
- } Ограничения:
  - ❖ Команда не может быть использована до создания поля: INIT, потому что в противном случае ей нечего будет резать
  - ❖ Команда не может быть использована до команды генерации поля (добавления на него Гауссов) либо команды чтения .bmp, потому что в противном случае поле тривиально и резать нечего

## ***WAVE min\_component\_size***

- } Выделяет связные компоненты и отбрасывает слишком мелкие
- } Где:

- ❖ `component_size` – целое число, показывающее, компоненты какого размера мы считаем шумом (мусором)

} Пример: WAVE 7

} Ограничения:

- ❖ Команда не может быть использована до команды генерации поля (добавления на него Гауссов) либо команды чтения `.bmp`, потому что в противном случае поле тривиально и искать компоненты связности смысла нет

### ***KMEANS amount\_of\_dots***

} Реализует алгоритм k-средних

} Где:

- ❖ `amount_of_dots` – целое число для алгоритма k-средних

} Пример: KMEANS 3

} Ограничения:

- ❖ Команда не может быть использована до команды WAVE, потому что в противном случае у пользователя на руках нет компонент, с которыми он хочет работать

### ***KMCORE amount\_of\_cores amount\_of\_dots\_in\_cores***

} Реализует алгоритм k-средних с ядрами

} Где:

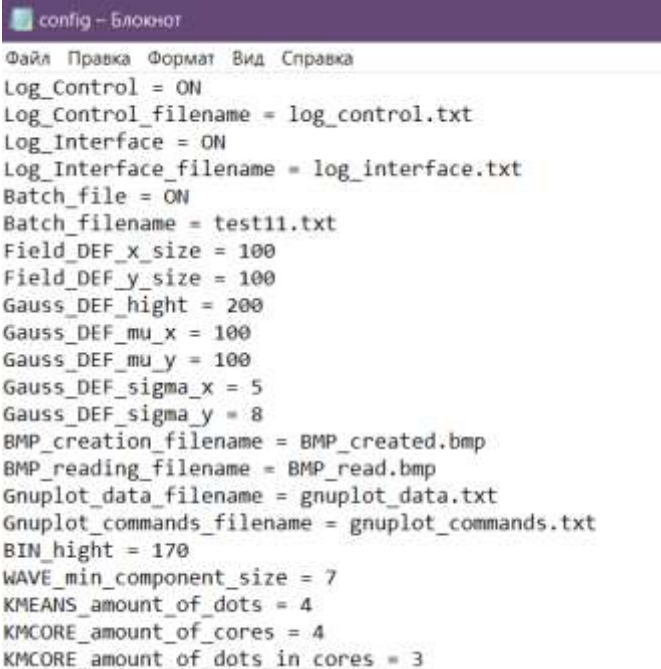
- ❖ `amount_of_dots` – целое число, количество точек в ядре
- ❖ `amount_of_dots_in_cores` – целое число, количество ядер

} Пример: KMCORE ?3

} Ограничения:

- ❖ Команда не может быть использована до команды WAVE, потому что в противном случае у пользователя на руках нет компонент, с которыми он хочет работать

## ПРИМЕР КОНФИГУРАЦИОННОГО ФАЙЛА



```
config - Блокнот
Файл Правка Формат Вид Справка
Log_Control = ON
Log_Control_filename = log_control.txt
Log_Interface = ON
Log_Interface_filename = log_interface.txt
Batch_file = ON
Batch_filename = test11.txt
Field_DEF_x_size = 100
Field_DEF_y_size = 100
Gauss_DEF_high = 200
Gauss_DEF_mu_x = 100
Gauss_DEF_mu_y = 100
Gauss_DEF_sigma_x = 5
Gauss_DEF_sigma_y = 8
BMP_creation_filename = BMP_created.bmp
BMP_reading_filename = BMP_read.bmp
Gnuplot_data_filename = gnuplot_data.txt
Gnuplot_commands_filename = gnuplot_commands.txt
BIN_high = 170
WAVE_min_component_size = 7
KMEANS_amount_of_dots = 4
KMCORE_amount_of_cores = 4
KMCORE_amount_of_dots_in_cores = 3
```

где

Log\_Control – вкл/выкл логирование (отчет о выполненных действиях) сервера

Log\_Control\_filename – имя файла для логирования сервера

Log\_Interface – вкл/выкл логирование интерфейса пользователя

Log\_Interface\_filename – имя файла для логирования интерфейса

Batch\_file – имя запускаемого файла по умолчанию вкл/выкл

Batch\_filename – имя запускаемого файла по умолчанию

Field\_DEF\_x\_size, Field\_DEF\_y\_size,

Gauss\_DEF\_high, Gauss\_DEF\_mu\_x, Gauss\_DEF\_mu\_y, Gauss\_DEF\_sigma\_x, Gauss\_DEF\_sigma\_y – значения по умолчанию для соответствующих полей поля и гауссиан, используются при нехватке параметров команды в запускаемом файле

BMP\_creation\_filename – имя создаваемого bmp файла

BMP\_reading\_filename – имя читаемого bmp файла

Gnuplot\_data\_filename – имя файла для данных для гнуплота (по этим данным строим график)

Gnuplot\_commands\_filename – имя файла непосредственно для команд гнуплота (собственно запускаемые команды, помещенные в файл)

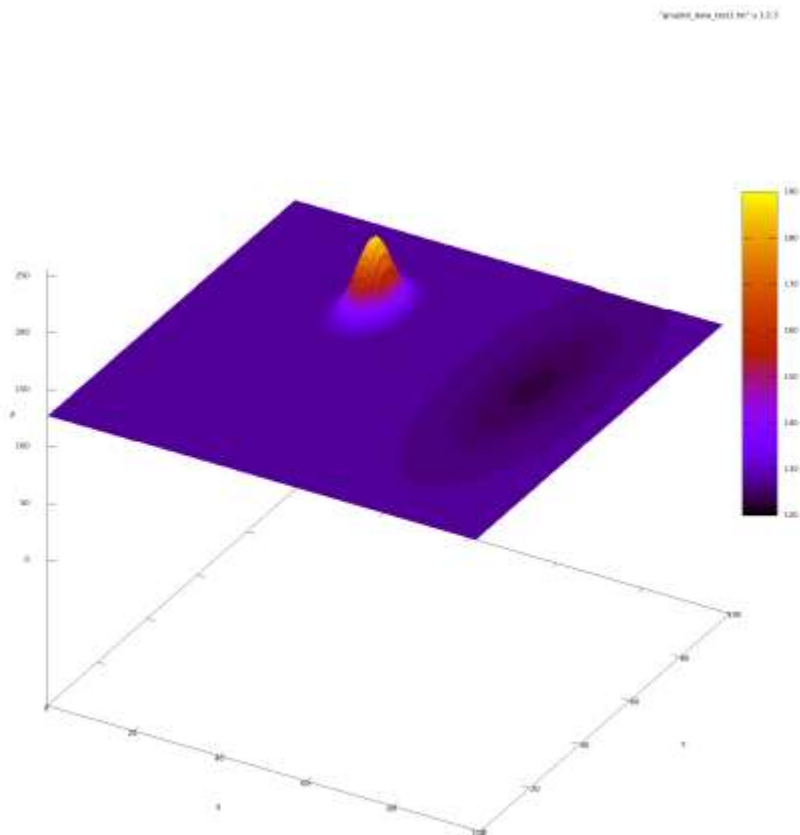
BIN\_high – высота среза по умолчанию

WAVE\_min\_component\_size, KMEANS\_amount\_of\_dots, KMCORE\_amount\_of\_cores, KMCORE\_amount\_of\_dots\_in\_cores – значения соответствующих полей k средних и k средних с ядрами по умолчанию (о полях читать подробнее в разделе выше с описанием команд)

## ПРИМЕРЫ ЗАПУСКА на различных входных файлах

1.

```
test1 - Блокнот
Файл Правка Формат Вид Справка
GSS 140 35 70 3 5
INIT 100 100
GSS 120 78 60 7 20
GEN
GNU gnuplot_data_test1.txt Gnuplot_test1.txt
```



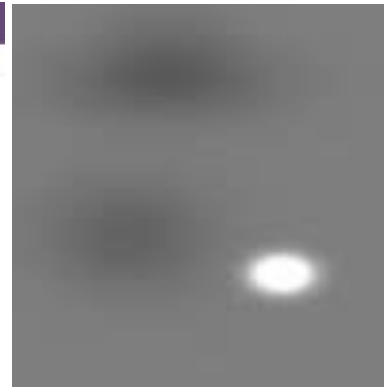
2.

```
test2 - Блокнот
Файл Правка Формат Вид Справка
```

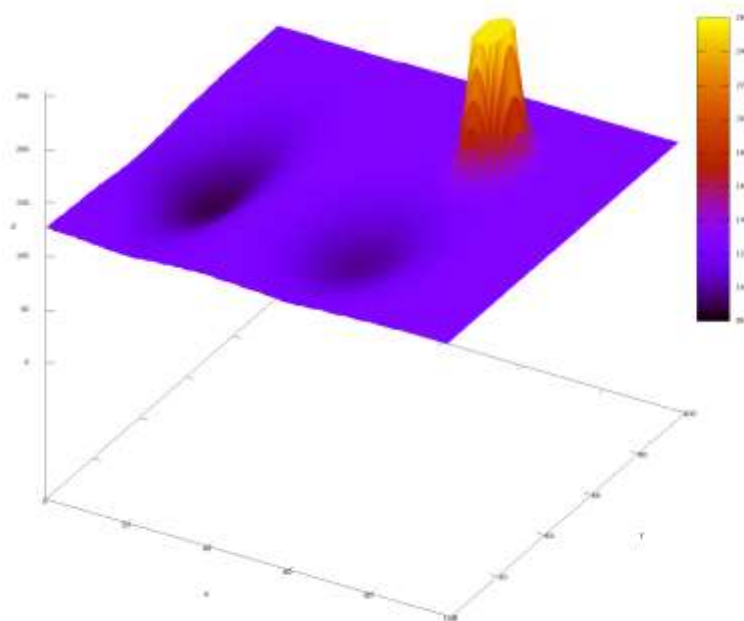
```
svetlana@LAPTOP-VUB0PJIN:~/Gauss-B$ ./prog
212-Chernyshova-Svetlana-Gauss.
We are going to make a field of gaussians.
Write the name of the configuration file :
config-1.txt
Succesfully opened file : config-1.txt
Interface logging enabled with file: log_control-1.txt
Interface logging enabled with file : log_interface-1.txt
How do you want to input the commands?
1: Concole
2: Text file
2
Please enter the file name :
test2.txt
Understood the command receiving place.
There are no more commands in the file. Everything is complete.
```

3.

```
test3 - Блокнот
Файл Правка Формат Вид Справка
GSS 250 70 70 3 5
GSS 10 60 30 10 15
INIT 100 100
GSS 10 20 40 7 20
GSS 90 10 40 8 12
GEN
GNU gnuplot_data_test3.txt Gnuplot_test3.txt
CRBMP bmp_created_test3.bmp
```

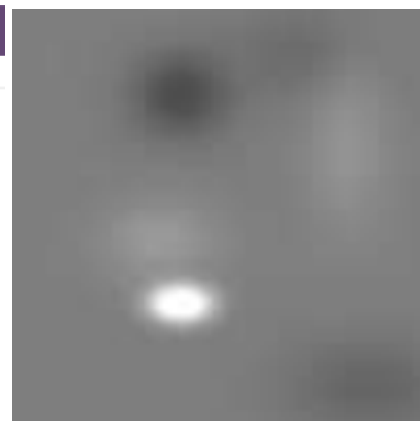


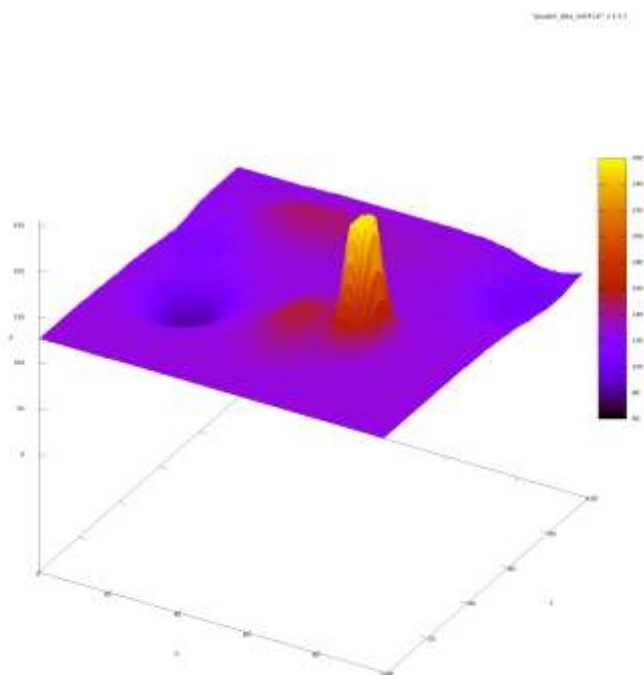
"bmp\_created\_test3.bmp" 2.2.2



4.

```
test4 - Блокнот
Файл Правка Формат Вид Справка
GSS 10 20 40 7 8
INIT 100 100
GSS 50 10 70 8 12
GSS 250 70 40 3 5
GSS 210 55 35 7 10
GSS 10 90 85 8 14
GSS 230 30 80 16 7
GEN
GNU gnuplot_data_test4.txt Gnuplot_test4.txt
CRBMP bmp_created_test4.bmp
```





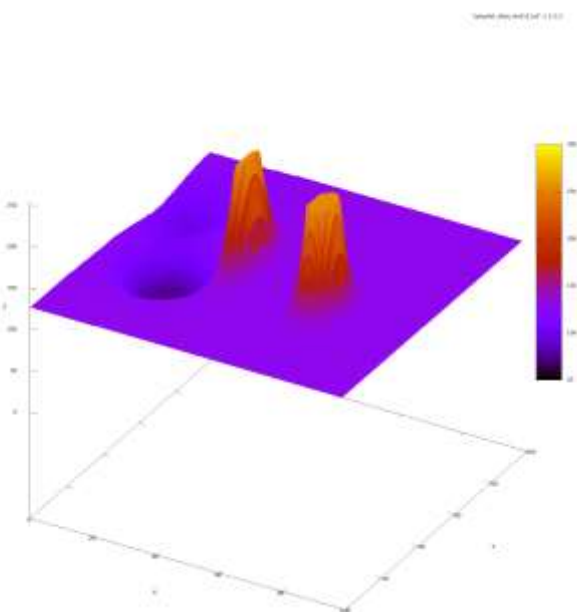
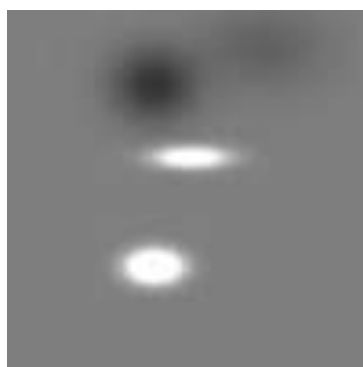
5.

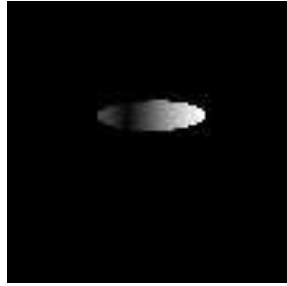
test12 - Блокнот

```

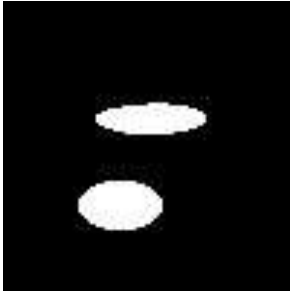
Файл  Правка  Формат  Вид  Справка
GSS 10 20 40 7 8
INIT 100 100
GSS 50 10 70 8 12
GSS 250 70 40 3 5
GSS 200 40 50 2 7
GEN
CRBMP bmp_created_test12.bmp
GNU gnuplot_data_test12.txt Gnuplot_test12.txt
BIN 130
WAVE 7
KMEANS 3
KMCORE 4 6

```





- срез покомпонентно



- все компоненты среза



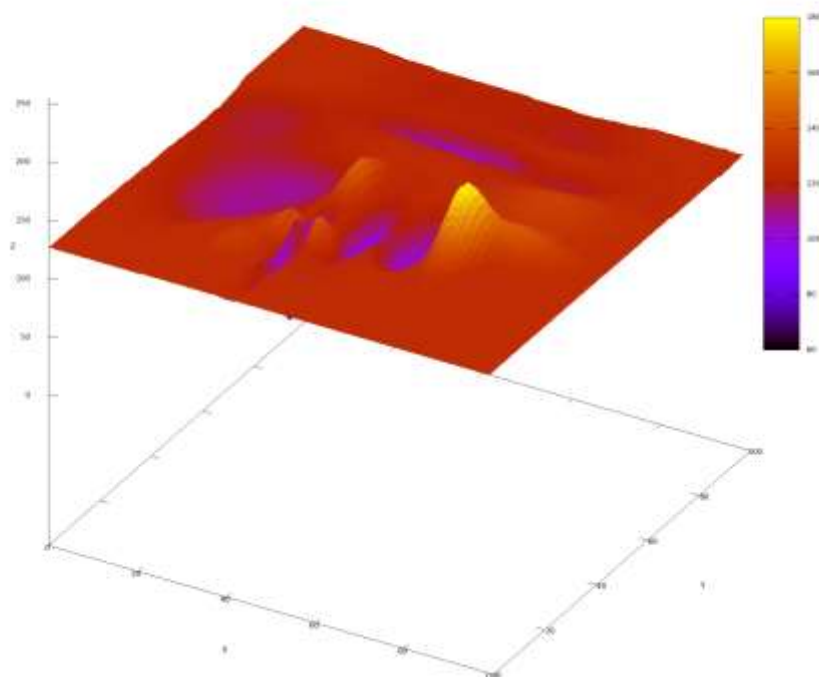
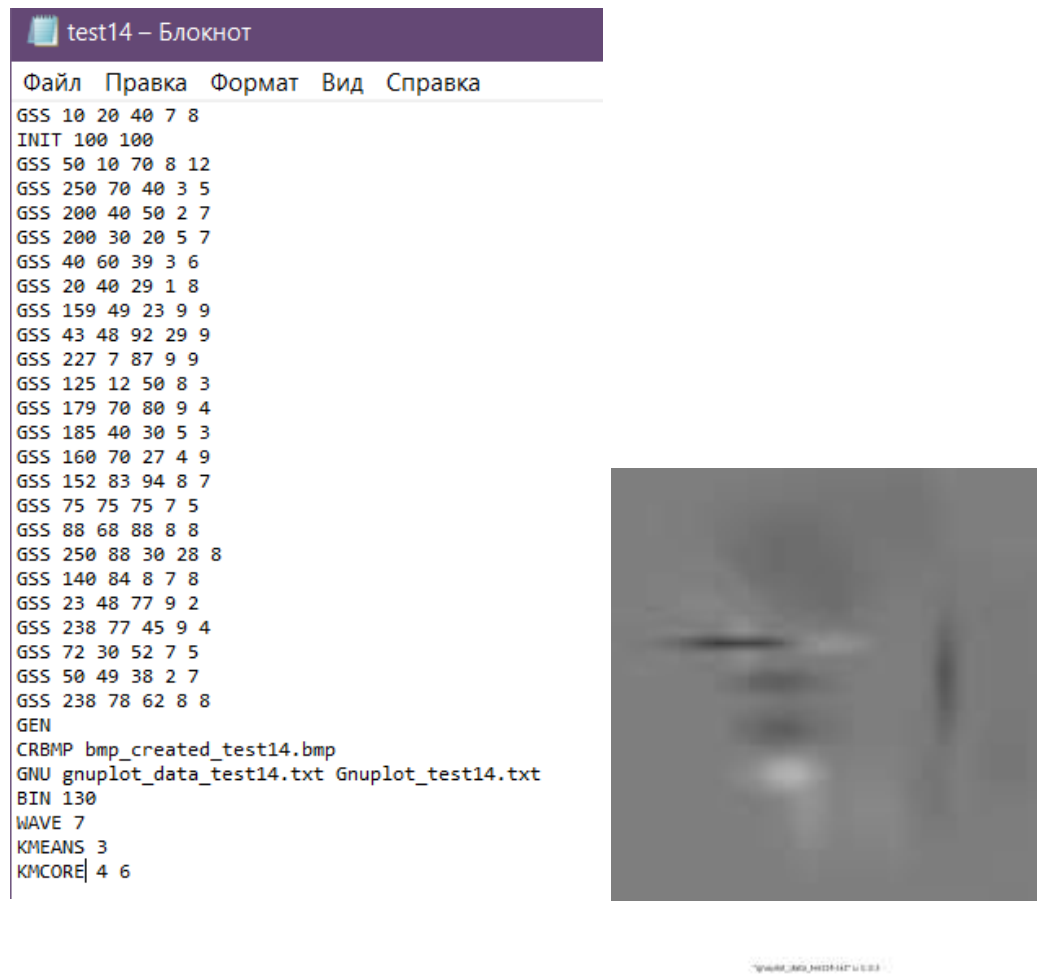
- кластеры k-means



- кластеры k-means с ядрами



6.





```
svetlana@LAPTOP-VUBQPJ1N:~/Gauss-8$ ./prog
212-Chernyshova-Svetlana-Gauss.
We are going to make a field of gaussians.
Write the name of the configuration file :
config-1.txt
Failed to open the configuration file : ccq♦config-1.txt
Succesfully opened file : ccq♦config-1.txt
How do you want to input the commands?
1: Concole
2: Text file
2
Please enter the file name :
test14.txt
Understood the command receiving place.
GSS
```

## ***ПРИМЕР работы конфигурационных файлов***

### ***Сервер:***

```
31.01.2025 02:12:19 : Class Control received control from Interface and received command GSS.
31.01.2025 02:12:19 : Gauss was successfully created.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command GEN.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command CRBMP.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command GNU.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command BIN.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command WAVE.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command KMEANS.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.

31.01.2025 02:12:19 : Class Control received control from Interface and received command KMCORES.
31.01.2025 02:12:19 : The command is completed, the report will be in the Interface.
The control is given back to the Interface.
```

### ***Интерфейс:***

```
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : The field generation is complete.
31.01.2025 02:12:19 : The interface has given control to the Control (class) and sent the command CRBMP to the Control. Waiting for an answer.
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : BMP file was successfully created.
31.01.2025 02:12:19 : The interface has given control to the Control (class) and sent the command GNU to the Control. Waiting for an answer.
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : Gnuplot file was successfully created.
31.01.2025 02:12:19 : The interface has given control to the Control (class) and sent the command BIN to the Control. Waiting for an answer.
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : The slice matrix is successfully created and added to the list.
31.01.2025 02:12:19 : The interface has given control to the Control (class) and sent the command WAVE to the Control. Waiting for an answer.
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : WAVE successfully completed.
31.01.2025 02:12:19 : The interface has given control to the Control (class) and sent the command KMEANS to the Control. Waiting for an answer.
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : Kmeans successfully completed.
31.01.2025 02:12:19 : The interface has given control to the Control (class) and sent the command KMCORE to the Control. Waiting for an answer.
31.01.2025 02:12:19 : The interface has returned the control and received a response from the Control.
31.01.2025 02:12:19 : REPORT : KMCORES successfully completed.
```