

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБРНАУКИ)

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ
И ОПТИКИ»
(СПбГУ ИТМО)

УТВЕРЖДАЮ
Ректор СПбГУ ИТМО,
докт. техн. наук, профессор
В. Н. Васильев

_____ 2008 г.

ПРОГРАММНОЕ СРЕДСТВО 3GENETIC

ПРОГРАММНЫЙ МОДУЛЬ CORE
ОПИСАНИЕ ПРОГРАММЫ

ЛИСТ УТВЕРЖДЕНИЯ

7.190.00001-01 13 02-ЛУ

Декан факультета «Информационные
технологии и программирование»
докт. техн. наук, профессор
_____ В. Г. Парфенов

Руководитель темы
заведующий кафедрой «Технологии программирования»,
докт. техн. наук, профессор
_____ А. А. Шалыто

Имя, И. подл.	Подп. и дата
Имя, И. дубл.	
Имя, И. дубл.	
Имя, И. дубл.	
Имя, И. дубл.	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБРНАУКИ)

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ
И ОПТИКИ»
(СПбГУ ИТМО)

УТВЕРЖДЕНО
7.190.00001-01 13 02-ЛУ

ПРОГРАММНОЕ СРЕДСТВО 3GENETIC

ПРОГРАММНЫЙ МОДУЛЬ CORE
ОПИСАНИЕ ПРОГРАММЫ

7.190.00001-01 13 02

Листов 33

Имя. N подл.	Подп. и дата	Взам. имя. N	Имя. N дубл.	Подп. и дата

АННОТАЦИЯ

В данном документе приводится описание модуля core программного средства 3GENETIC, представляющего собой графическую оболочку генетического алгоритма, а также реализующего все базовую функциональность ядра.

СОДЕРЖАНИЕ

Содержание.....	3
1. Общие сведения	5
2. Функциональное назначение	6
3. Описание логической структуры.....	7
3.1. Класс ChooseDialog	7
3.1.1. Открытые члены.....	7
3.1.2. Конструктор(ы)	7
3.1.3. Методы.....	7
3.2. Класс ConfigDialog.....	7
3.2.1. Открытые члены.....	8
3.2.2. Конструктор(ы)	8
3.2.3. Методы.....	8
3.3. Класс FrameCreator	8
3.3.1. Открытые члены.....	8
3.3.2. Конструктор(ы)	8
3.3.3. Методы.....	8
3.4. Класс PauseAction	9
3.4.1. Открытые члены.....	9
3.4.2. Конструктор(ы)	9
3.4.3. Методы.....	9
3.5. Класс ResumeAction	9
3.5.1. Открытые члены.....	10
3.5.2. Методы.....	10
3.6. Класс TimerListener	10
3.6.1. Открытые члены.....	10
3.6.2. Методы.....	11
3.7. Класс MainFrame	11
3.7.1. Открытые члены.....	11
3.7.2. Конструктор(ы)	11
3.7.3. Методы.....	11
3.8. Класс ChooseGAAction	12
3.8.1. Открытые члены.....	12
3.8.2. Конструктор(ы)	12
3.8.3. Методы.....	12
3.9. Класс ChooseIndividualAction	12
3.9.1. Открытые члены.....	12
3.9.2. Конструктор(ы)	13
3.9.3. Методы.....	13
3.10. Класс ExitAction	13
3.10.1. Открытые члены.....	13
3.10.2. Конструктор(ы)	13
3.10.3. Методы.....	13
3.11. Класс SaveGenerationAction	14
3.11.1. Открытые члены.....	14
3.11.2. Конструктор(ы)	14
3.11.3. Методы.....	14
3.12. Класс SavePictureAction.....	14
3.12.1. Открытые члены.....	15
3.12.2. Конструктор(ы)	15
3.12.3. Методы.....	15

3.13.	Класс ShowGenerationAction.....	15
3.13.1.	Открытые члены.....	15
3.13.2.	Конструктор(ы).....	15
3.13.3.	Методы.....	16
3.14.	Класс MainMenu.....	16
3.14.1.	Конструктор(ы).....	16
3.15.	Класс IndividualDialog.....	16
3.15.1.	Открытые члены.....	16
3.15.2.	Конструктор(ы).....	17
3.15.3.	Методы.....	17
3.16.	Класс ShowTableModel.....	17
3.16.1.	Открытые члены.....	17
3.16.2.	Конструктор(ы).....	17
3.16.3.	Методы.....	18
3.17.	Класс PluginLoader.....	18
3.17.1.	Открытые члены.....	18
3.17.2.	Статические открытые данные.....	19
3.17.3.	Конструктор(ы).....	19
3.17.4.	Методы.....	19
3.18.	Класс Main.....	19
3.18.1.	Конструктор(ы).....	20
3.18.2.	Методы.....	20
3.19.	Класс GARunner.....	20
3.19.1.	Открытые члены.....	20
3.19.2.	Статические открытые данные.....	20
3.19.3.	Конструктор(ы).....	20
3.19.4.	Методы.....	21
3.20.	Класс SelectedPlugin.....	21
3.20.1.	Конструктор(ы).....	21
3.20.2.	Методы.....	21
3.21.	Класс RunnableLock.....	22
3.21.1.	Открытые члены.....	22
3.21.2.	Конструктор(ы).....	22
3.21.3.	Методы.....	22
4.	Используемые технические средства.....	23
5.	Вызов и загрузка.....	24
6.	Входные данные.....	26
7.	Выходные данные.....	30

1. ОБЩИЕ СВЕДЕНИЯ

Программа построения автоматов управления системами со сложным поведением с помощью генетического программирования, написана на языке программирования *Java*.

Для нормального функционирования данной программы необходимо, чтобы на персональной ЭВМ была установлена одна из следующих операционных систем:

- *Microsoft Windows 2000 Professional* (русская и английская версии), с установленным *Service Pack 4*;
- *Windows XP Professional* (русская и английская версии), с установленным *Service Pack 3*.

Также необходимо, чтобы на персональной ЭВМ была установлена среда разработки программного обеспечения на языке *Java*, версия не ниже *jdk1.6.0_xx*;

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Описываемая программа предназначена для построения с помощью генетического программирования конечных автоматов, управляющих системами со сложным поведением.

Ядро программы позволяет просматривать и сохранять в виде файлов графики зависимости значений некоторых функций (например, максимального, минимального и среднего значения функции приспособленности особей поколения) от номера поколения. Кроме этого поддерживается возможность визуализации особей и их сохранения в текстовом формате. Также ядро программы поддерживает подключение плагинов налету (во время работы программы).

В качестве подключаемых модулей выступают:

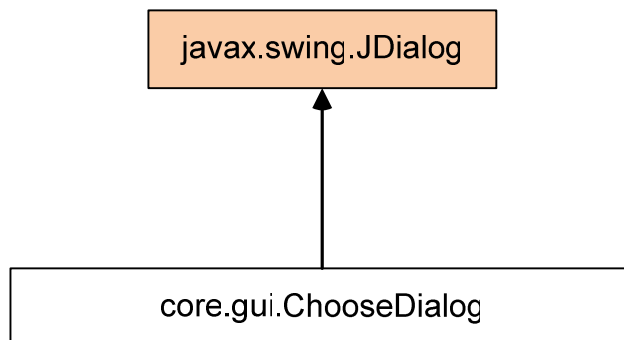
- функции, графики которых строятся ядром программы;
- реализации различных генетических алгоритмов;
- реализации различных представлений особей и операций скрещивания и мутации для генетических алгоритмов;
- визуализаторы особей и поведения задаваемых ими автоматов (они зависят от конкретной задачи и от конкретного представления особи).

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1.Класс ChooseDialog

Этот класс описывает окно для выбора подключаемого модуля.

Граф наследования:



3.1.1. Открытые члены

- valueChanged(ListSelectionEvent): void
- actionPerformed(ActionEvent): void

3.1.2. Конструктор(ы)

- ChooseDialog(Frame f, PluginLoader load, int index) – создает окно для выбора части плагина.

3.1.3. Методы

- valueChanged(ListSelectionEvent e): void - обрабатывает событие выбора строчек, устанавливает комментарий, в зависимости от выбранных строчек
 - e – событие изменения в диалоговом окне.
- actionPerformed(ActionEvent e): void - обрабатывает события нажатия на кнопки.
 - e – событие изменения в диалоговом окне.

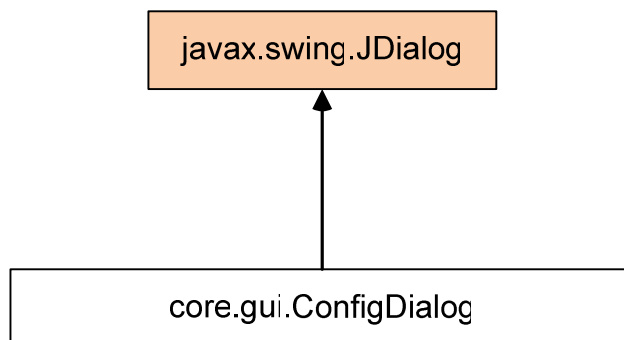
Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\chooser\ChooseDialog.java

3.2.Класс ConfigDialog

Этот класс описывает окно для изменения настроек встраиваемого модуля.

Граф наследования:



3.2.1. Открытые члены

- actionPerformed(ActionEvent): void - обрабатывает события нажатия на кнопки.

3.2.2. Конструктор(ы)

- ConfigDialog(JDialog, Properties) - создает окно для изменения настроек части плагина.

3.2.3. Методы

- actionPerformed(ActionEvent e): void - обрабатывает события нажатия на кнопки.

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\chooser\ConfigDialog.java

3.3. Класс FrameCreator

Этот класс создает главное окно приложения.

Классы наследники: отсутствуют.

3.3.1. Открытые члены

- run(): void - создает главное окно и снимает блокировку.

3.3.2. Конструктор(ы)

FrameCreator(GARunner, PluginLoader, Lock).

3.3.3. Методы

- run(): void - создает главное окно.

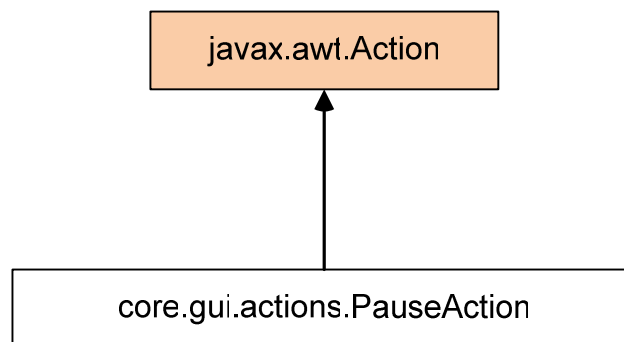
Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\FramеCreator.java

3.4. Класс PauseAction

Этот класс описывает действия, выполняемые для постановки фабрики на паузу.

Граф наследования:



3.4.1. Открытые члены

- setNewAction(Action): void - устанавливает новое действие.
- getLock(): Lock - возвращает блокировку.
- actionPerformed(ActionEvent): void - обрабатывает нажатие на кнопку, ставит фабрику на паузу.

3.4.2. Конструктор(ы)

- PauseAction(GARunner, String, String) - запоминает параметры, присваивает имя и создает блокировку.

3.4.3. Методы

- setNewAction(Action a): void - устанавливает новое действие.
 - a – новое действие
- getLock(): Lock - возвращает блокировку.
- actionPerformed(ActionEvent): void - обрабатывает нажатие на кнопку, ставит фабрику на паузу.
 - e – событие нажатия на кнопку.

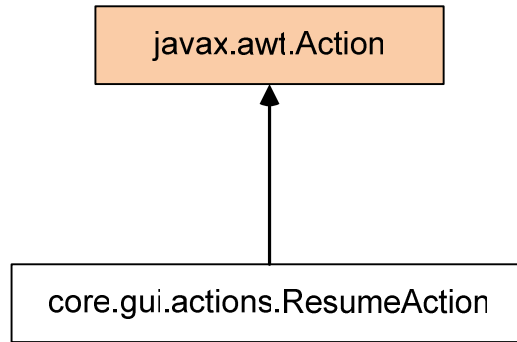
Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\actions\PauseAction.java

3.5. Класс ResumeAction

Этот класс описывает действия, выполняемые для снятия фабрики с паузы.

Граф наследования:



3.5.1. Открытые члены

- setNewAction(Action): void
- getLock(): Lock
- actionPerformed(ActionEvent): void

3.5.2. Методы

- setNewAction(Action): void - устанавливает новое действие.
 - a – новое действие
- getLock(): Lock - возвращает блокировку.
- actionPerformed(ActionEvent): void - обрабатывает нажатие на кнопку, ставит фабрику на паузу.
 - e – событие нажатия на кнопку.

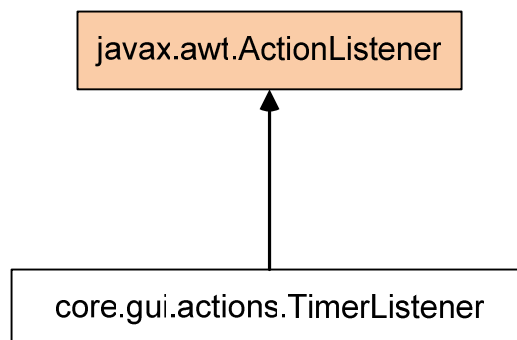
Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\actions\ResumeAction.java

3.6. Класс TimerListener

Этот класс перерисовывает заданное окно по событию.

Граф наследования:



3.6.1. Открытые члены

- actionPerformed(ActionEvent e): void

3.6.2. Методы

- `actionPerformed(ActionEvent)`: `void` – перерисовывает окно.
 - `e` – событие срабатывания таймера

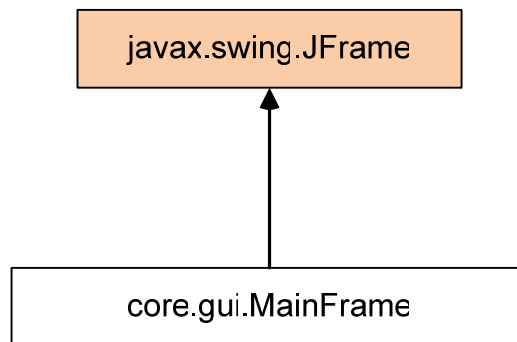
Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\actions\TimerListener.java`
-

3.7. Класс MainFrame

Этот класс описывает главное окно, предоставляет возможность сохранять изображения вкладок и пересоздавать их.

Граф наследования:



3.7.1. Открытые члены

- `imagePanel()`: `BufferedImage`
- `changeGraphicPanels()`: `void`

3.7.2. Конструктор(ы)

- `MainFrame(String, GARunner, PluginLoader)` - создает главное окно приложения и подготавливает все его компоненты.

3.7.3. Методы

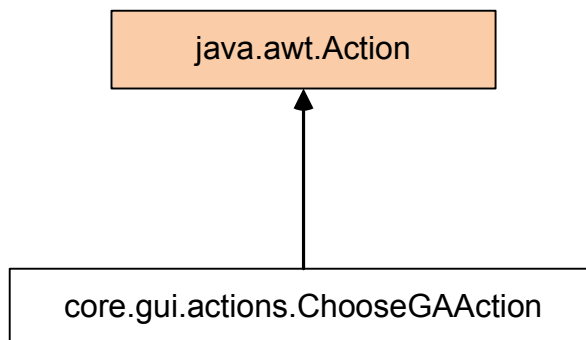
- `imagePanel()`: `BufferedImage` - возвращает изображение одной из вкладок.
- `changeGraphicPanels()`: `void` - переупорядочивает вкладки графиков главного окна.

Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\main\MainFrame.java`

3.8.Класс ChooseGAAction

Этот класс описывает элемент меню, который обрабатывает выбор генетического алгоритма. Граф наследования:



3.8.1. Открытые члены

- actionPerformed(ActionEvent): void

3.8.2. Конструктор(ы)

- ChooseGAAction(Frame, PluginLoader) - создает элемент меню, присваивает имя и запоминает параметры вызова диалога.

3.8.3. Методы

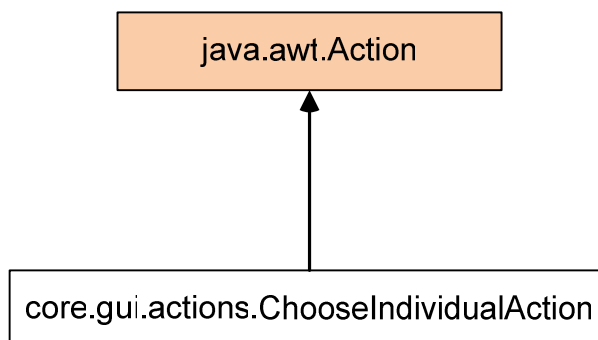
- actionPerformed(ActionEvent e): void – отвечает на действие, выбор пункта меню, открытием диалога выбора генетического алгоритма. Выводит предупреждение, если смена генетического алгоритма завершилась неуспешно.
 - e – событие нажатия на кнопку

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\main\menu\actions\ChooseGAAction.java

3.9.Класс ChooseIndividualAction

Этот класс описывает элемент меню, который обрабатывает выбор представления особи. Граф наследования:



3.9.1. Открытые члены

- actionPerformed(ActionEvent): void

3.9.2. Конструктор(ы)

- ChooseIndividualAction(Frame, PluginLoader) - создает элемент меню, присваивает имя и запоминает параметры вызова диалога.
-

3.9.3. Методы

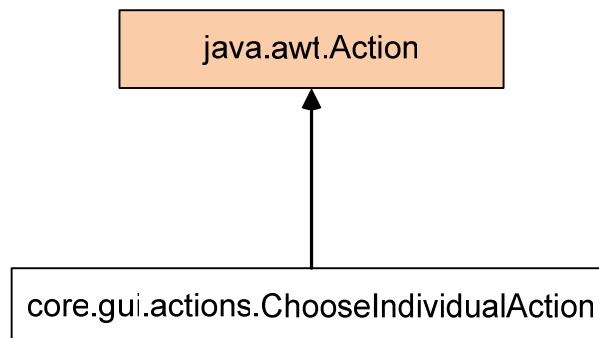
- actionPerformed(ActionEvent): void – отвечает на действие "выбор пункта меню", открытием диалога выбора представления особи. Если представление было изменено успешно, то будут изменены графики главного окна, иначе будет выведено предупреждение.
 - e – событие нажатия на кнопку
-

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\main\menu\actions\ChooseIndividualAction.java

3.10. Класс ExitAction

Этот класс описывает элемент меню, который обрабатывает выход из программы.
Граф наследования:



3.10.1. Открытые члены

- actionPerformed(ActionEvent): void
-

3.10.2. Конструктор(ы)

- ExitAction(Frame, PluginLoader) - создает элемент меню, указывает короткое описание.
-

3.10.3. Методы

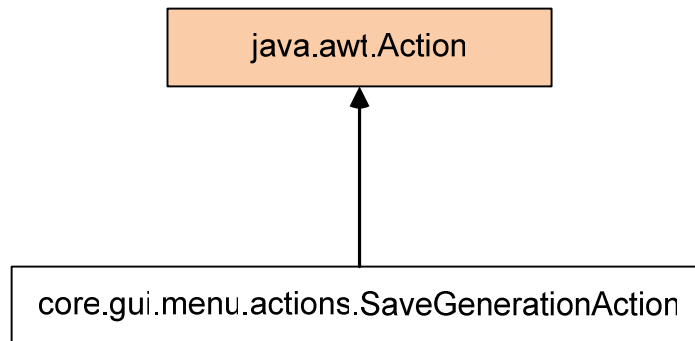
- actionPerformed(ActionEvent e): void – выводит диалог подтверждения выхода, при положительном ответе завершает работу программы.
 - e – событие нажатия на кнопку
-

Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\main\menu\actions\ExitAction.java`

3.11. Класс SaveGenerationAction

Этот класс описывает элемент меню, который обрабатывает сохранение поколения.
Граф наследования:



Классы наследники: отсутствуют.

3.11.1. Открытые члены

- `actionPerformed(java.awt.event.ActionEvent): void`

3.11.2. Конструктор(ы)

- `SaveGenerationAction(GARunner, int)` - создает элемент меню, присваивает ему имя, сохраняет фабрику

3.11.3. Методы

- `actionPerformed(ActionEvent e): void` – отвечает на действие, выбор пункта меню. Сохраняет 50 особей в папку "results", все файлы из папки до сохранения будут удалены. При отсутствии особей будет выведено предупреждение.
 - `e` – событие нажатия на кнопку

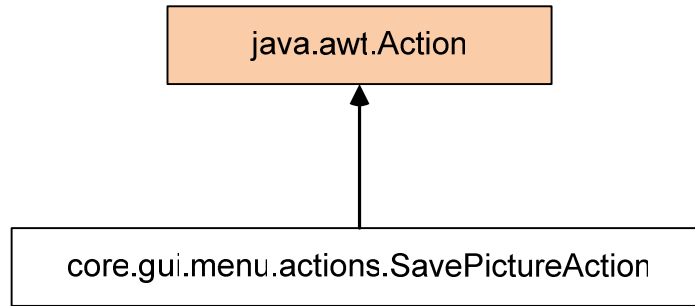
Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\main\menu\actions\SaveGenerationAction.java`

3.12. Класс SavePictureAction

Этот класс описывает элемент меню, который обрабатывает сохранение изображения открытой вкладки.

Граф наследования:



3.12.1. Открытые члены

- actionPerformed(ActionEvent): void

3.12.2. Конструктор(ы)

- SavePictureAction(GARunner, int) - создает элемент меню, присваивает ему имя и подготавливает диалог сохранения файла.

3.12.3. Методы

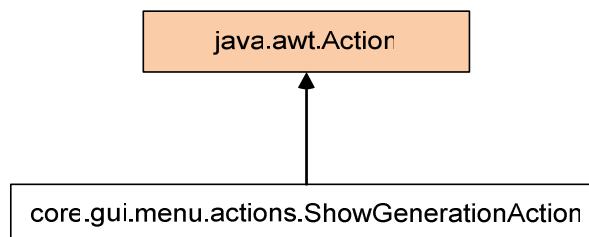
- actionPerformed(ActionEvent e): void – отображает диалог выбора файла и сохраняет изображение вкладки в выбранный файл.
 - e – событие нажатия на кнопку

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\main\menu\actions\SavePictureAction.java

3.13. Класс ShowGenerationAction

Этот класс описывает элемент меню, который обрабатывает отображение текущей популяции. Граф наследования:



3.13.1. Открытые члены

- actionPerformed(ActionEvent): void

3.13.2. Конструктор(ы)

- ShowGenerationAction(GARunner, int) - создает элемент меню, присваивает имя и запоминает параметры вызова диалога.
-

3.13.3. Методы

- `actionPerformed(ActionEvent e): void` – отображает диалоговое окно с особями, соответствующими индексу.
 - `e` – событие нажатия на кнопку

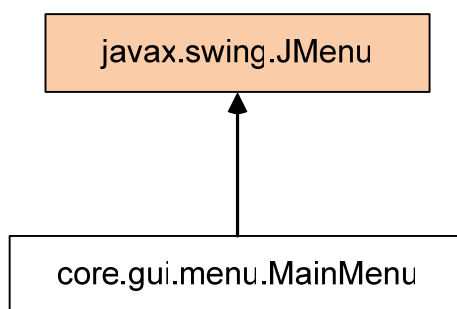
Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\main\menu\actions\ShowGenerationAction.java`

3.14. Класс MainMenu

Этот класс описывает меню главного окна.

Граф наследования:



3.14.1. Конструктор(ы)

- `MainMenu(GARunner, PluginLoader, MainFrame)` - создает меню главного окна и настраивает его компоненты.

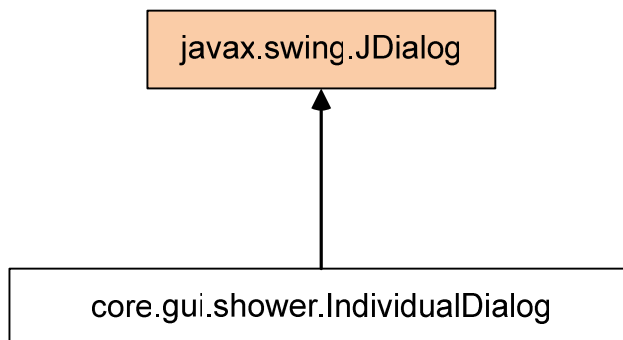
Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\main\menu\MainMenu.java`

3.15. Класс IndividualDialog

Этот класс описывает диалоговое окно отображения особей.

Граф наследования:



3.15.1. Открытые члены

- `valueChanged(ListSelectionEvent): void`
- `actionPerformed(ActionEvent): void`
- `getSelectedIndex(): int`

3.15.2. Конструктор(ы)

- IndividualDialog(List, List, PluginLoader) - отображает окно с особями.

3.15.3. Методы

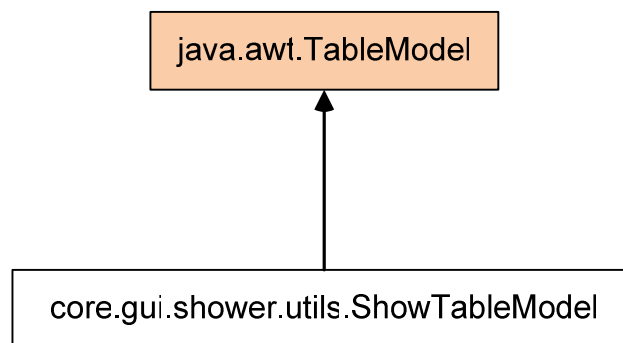
- valueChanged(ListSelectionEvent e): void - из выделенных строчек выбирает первую.
 - e – событие изменения окна.
- actionPerformed(ActionEvent e): void - закрывает окно.
 - e – событие закрытия окна
- getSelectedIndex(): int - возвращает номер выделенной строки.

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\gui\shower\IndividualDialog.java

3.16. Класс ShowTableModel

Этот класс описывает таблицу, для отображения группы особей.
Граф наследования:



Классы наследники: отсутствуют.

3.16.1. Открытые члены

- getColumnCount(): int
- getColumnName(int): String
- getRowCount(): int
- isCellEditable(int, int): boolean
- getColumnClass(int): Class
- setValueAt(Object, int, int): void
- getValueAt(int, int): Object

3.16.2. Конструктор(ы)

- ShowTableModel(List, List) - сохраняет особей, номера поколений и названия колонок.
-

3.16.3. Методы

- `getColumnCount(): int` - возвращает количество колонок.
- `getColumnName(int index): String` - возвращает название колонки по номеру.
 - `index` – номер колонки
- `getRowCount(): int` - возвращает количество строчек.
- `isCellEditable(int row, int column): boolean` - возвращает `true`, если ячейку можно редактировать.
 - `row` – номер строки
 - `column` – номер столбца
- `getColumnClass(int column): Class` - возвращает класс колонки по номеру.
 - `column` – номер столбца
- `setValueAt(Object, int row, int column): void` - устанавливает выделение ячейки.
 - `value` – записываемое значение
 - `row` – номер строки
 - `column` – номер столбца
- `getValueAt(int row, int column): Object` - возвращает значение ячейки таблицы.
 - `row` – номер строки
 - `column` – номер столбца

Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\gui\shower\utils>ShowTableModel.java`

3.17. Класс PluginLoader

Этот класс загружает плагины (генетические алгоритмы) из указанных директорий и предоставляет к ним доступ.

Классы наследники: отсутствуют.

3.17.1. Открытые члены

- `getGA(int, IndividualFactory): GA`
- `getIndividualFactory(int): IndividualFactory`
- `getEmulator(int, int, Visualizable): Frame`
- `getName(int, int): String`
- `getComment(int, int): String`
- `getCount(int): int`
- `getCountEmulator(int): int`
- `getIndividualMax(int): double`
- `getIndividualCountSigns(int): int`
- `getProperties(int, int): Properties`
- `getFunctors(): Functor[]`
- `getFunctorTitle(int): String`

3.17.2. Статические открытые данные

- INDEX_GA: int - индекс, определяющий генетические алгоритмы.
 - INDEX_INDIVIDUAL: int - индекс, определяющий представление особи.
-

3.17.3. Конструктор(ы)

- PluginLoader(File, File, File, File) - этот конструктор загружает и запоминает все плагины из указанных директорий.
-

3.17.4. Методы

- getGA(int index, IndividualFactory factory): GA - возвращает класс, работающий с популяцией.
 - index – номер алгоритма
 - factory – фабрика
 - getIndividualFactory(int index): IndividualFactory - возвращает класс указанного плагина, создающий произвольных особей.
 - index – номер фабрики
 - getName(int index, int number): String - возвращает название указанного плагина.
 - index – тип модуля
 - number – номер модуля
 - getComment(int index, int number): String - возвращает комментарий к указанному плагину.
 - index – тип модуля
 - number – номер модуля
 - getCount(int index): int - возвращает количество указанных частей плагина.
 - index – тип модуля
 - getIndividualMax(int index): double - возвращает максимальное значение для графика.
 - index – номер особи
 - getIndividualCountSigns(int index): int - возвращает количество отображаемых знаков фитнес-функции.
 - index – номер особи
 - getProperties(int index, int number): Properties - возвращает настройки указанной части плагина.
 - index – тип модуля
 - number – номер модуля
 - getFunctors(): Functor[] - возвращает функцию графика.
 - getFunctorTitle(int index): String - возвращает название графика.
 - index – номер функтора
-

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\loader\PluginLoader.java

3.18. Класс Main

Этот класс запускает приложение.

Классы наследники: отсутствуют.

3.18.1. Конструктор(ы)

- Main() – конструктор по умолчанию.
-

3.18.2. Методы

- getPath(String file): File - возвращает директорию по обозначению.
 - file – имя файла
 - main(String[] args): void - запускает приложение. Загружает плагины, запускает генетический алгоритм, создает окно.
-

Объявления и описания членов класса находятся в файле:

- src\laboratory\core>Main.java

3.19. Класс GARunner

Этот класс-фабрика содержит генетический алгоритм и управляет производством новых поколений.

Классы наследники: отсутствуют.

3.19.1. Открытые члены

- run(): void - этот метод запускает генетический алгоритм и передает новые поколения функциям.
- tryToInterrupt(): void - прекращает работу генетического алгоритма, после завершения обработки текущего поколения.
- pause(): void - устанавливает паузу.
- resume(): void - снимает паузу.
- setLock(Lock lock): void - устанавливает новую блокировку.
 - lock – устанавливаемая блокировка
- getFuncutors(): Functor[] - возвращает графики функций.
- getIndividuals(int index): List<Individual> - возвращает список особей по индексу.
 - index – номер поколения
- getNumberGen(): List<Integer> - возвращает номера поколений, в которых появлялись лучшие особи.
- getCurrentGeneration(): int - возвращает номер текущего поколения.
- setGA(GA): int - устанавливает новый генетический алгоритм.

2.

3.19.2. Статические открытые данные

- CURRENT_GENERATION_INDEX: int – номер текущего поколения.
 - BEST_INDIVIDUAL_INDEX: int – номер лучшей особи.
-

3.19.3. Конструктор(ы)

- GARunner(GA, Functor[]) - запоминает генетический алгоритм и графики функций
-

3.19.4. Методы

- `run(): void` - этот метод запускает генетический алгоритм и передает новые поколения функциям.
- `tryToInterrupt(): void` - прекращает работу генетического алгоритма, после завершения обработки текущего поколения.
- `pause(): void` - устанавливает паузу.
- `resume(): void` - снимает паузу.
- `setLock(Lock lock): void` - устанавливает новую блокировку.
 - `lock` – устанавливаемая блокировка;
- `getFunctors(): Functor[]` - возвращает графики функций.
- `getIndividuals(int index): List<Individual>` - возвращает список особей по номеру поколения.
 - `index` – номер поколения
- `getNumberGen(): List<Integer>` - возвращает номера поколений, в которых появлялись лучшие особи.
- `getCurrentGeneration(): int` - возвращает номер текущего поколения.
- `setGA(GA ga): int` - устанавливает новый генетический алгоритм.
 - `ga` – устанавливаемый алгоритм.

Объявления и описания членов класса находятся в файле:

- `src\laboratory\core\runner\GARunner.java`

3.20. Класс SelectedPlugin

Этот класс хранит информацию о выбранном подключаемом модуле.

Классы наследники: отсутствуют.

3.20.1. Конструктор(ы)

- `SelectedPlugin()` – конструктор по умолчанию.
-

3.20.2. Методы

- `setIndexGA(int i): void` - задаёт генетический алгоритм.
 - `i` – номер алгоритма
 - `getIndexGA(): int` - возвращает номер выбранного генетического алгоритма.
 - `reDoGA(): void` - восстанавливает прежний номер используемого генетического алгоритма.
 - `setIndividual(int i): void` - устанавливает номер представления особи.
 - `i` – номер представления
 - `getIndexIndividual(): int` - возвращает номер представления особи.
 - `reDoIndividual(): void` - восстанавливает прежний номер представления особи.
 - `setIndexEmulator(int): void` - устанавливает номер визуализатора.
 - `i` – номер алгоритма
 - `getIndexEmulator(): int` - возвращает номер визуализатора.
 - `reDoEmulator(): void` - восстанавливает прежний номер визуализатора.
 - `getIndex(int): int` - возвращает номер части плагина, в зависимости от запрошенной части.
 - `setIndex(int, int): void` - устанавливает номер части плагина.
-

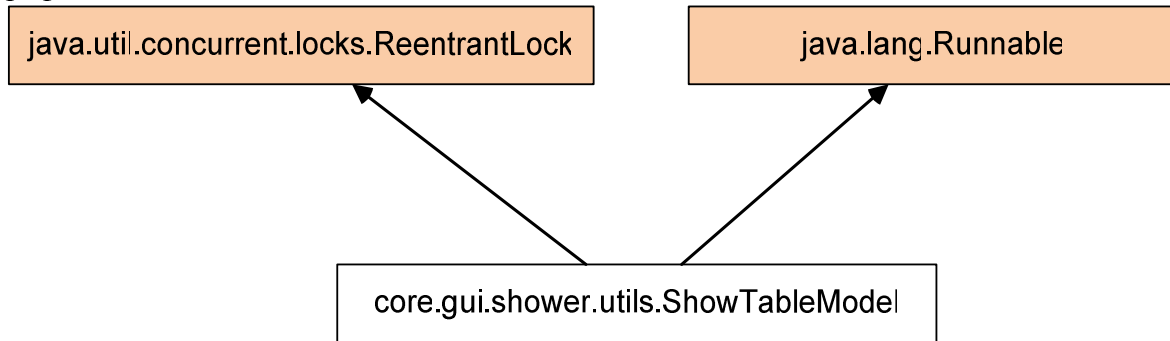
Объявления и описания членов класса находятся в файле:

- src\laboratory\core\SelectedPlugin.java

3.21. Класс RunnableLock

Этот класс хранит информацию о выбранном подключаемом модуле.

Граф наследования:



3.21.1. Открытые члены

- `run(): void`.

3.21.2. Конструктор(ы)

- `RunnableLock()` – конструктор по умолчанию.

3.21.3. Методы

- `run(): void` – устанавливает блокировку.

Объявления и описания членов класса находятся в файле:

- src\laboratory\core\util\RunnableLock.java

-

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для нормального функционирования программы автоматического построения с помощью генетического программирования конечных автоматов, управляющих системами со сложным поведением, на персональной ЭВМ, необходимо, чтобы аппаратное обеспечение персональной ЭВМ удовлетворяло следующим требованиям:

- процессор *Intel Pentium IV* или совместимый;
- тактовая частота процессора 2ГГц, не менее;
- оперативная память 1024 МВ, не менее;
- дисковый накопитель объемом 1 GB, не менее;
- отображающее устройство (монитор) с поддержкой разрешения 1024x768;
- устройства ввода клавиатура и мышь (трекбол, тачпад);

5. ВЫЗОВ И ЗАГРУЗКА

- public Choosedialog.valueChanged(ListSelectionEvent): void
- public Choosedialog.actionPerformed(ActionEvent): void
- public ConfigDialog.actionPerformed(ActionEvent): void
- public FrameCreator.run():void
- public PauseAction.setNewAction(Action): void
- public PauseAction.getLock(): Lock
- public PauseAction.actionPerformed(ActionEvent): void
- public ResumeAction.setNewAction(Action): void
- public ResumeAction.getLock(): Lock
- public ResumeAction.actionPerformed(ActionEvent): void
- public TimerListener.actionPerformed(ActionEvent e): void
- public MainFrame.imagePanel(): BufferedImage
- public MainFrame.changeGraphicPanels(): void
- public ChooseGAAction.actionPerformed(ActioEvent e):void
- public ChooseIndividualAction.actionPerformed(ActioEvent e):void
- public ExitAction.actionPerformed(ActioEvent e):void
- public SaveGenerationAction.actionPerformed(ActioEvent e):void
- public ShowGenerationAction.actionPerformed(ActioEvent e):void
- public SavePictureAction.actionPerformed(ActioEvent e):void
- public IndividualDialog.valueChanged(ListSelectionEvent): void
- public IndividualDialog.actionPerformed(ActionEvent): void
- public IndividualDialog.getSelectedIndex(): int
- public ShowTableModel.getColumnCount(): int
- public ShowTableModel.getColumnName(int): String
- public ShowTableModel.getRowCount(): int
- public ShowTableModel.isCellEditable(int, int): boolean
- public ShowTableModel.getColumnClass(int): Class
- public ShowTableModel.setValueAt(Object, int, int): void
- public ShowTableModel.getValueAt(int, int): Object
- public PluginLoader.getGA(int, IndividualFactory): GA
- public PluginLoader.getIndividualFactory(int): IndividualFactory
- public PluginLoader.getEmulator(int, int, Visualizable): Frame
- public PluginLoader.getName(int, int): String
- public PluginLoader.getComment(int, int): String
- public PluginLoader.getCount(int): int
- public PluginLoader.getCountEmulator(int): int
- public PluginLoader.getIndividualMax(int): double
- public PluginLoader.getIndividualCountSigns(int): int
- public PluginLoader.getProperties(int, int): Properties
- public PluginLoader.getFuncutors(): Functor[]
- public PluginLoader.getFuncutorTitle(int): String
- public Main.main():void
- public GARunner.run(): void
- public GARunner.tryToInterrupt(): void
- public GARunner.pause(): void
- public GARunner.resume(): void

- `public GARunner.setLock(Lock lock): void`
- `public GARunner.getFuncutors(): Functor[]`
- `public GARunner.getIndividuals(int index): List<Individual>`
- `public GARunner.getNumberGen(): List<Integer>`
- `public GARunner.getCurrentGeneration(): int`
- `public GARunner.setGA(GA): int`
- `public RunnableLock.run():void`

6. ВХОДНЫЕ ДАННЫЕ

- ChooseDialog.valueChanged(ListSelectionEvent e): void
 - e – событие изменения в диалоговом окне.
- ChooseDialog.actionPerformed(ActionEvent e): void
 - e – событие изменения в диалоговом окне.

- Config.actionPerformed(ActionEvent e): void
 - e – событие изменения в диалоговом окне.

- FrameCreator.run() :void
 - нет

- PauseAction.setNewAction(Action a): void
 - a – новое действие
- PauseAction.getLock(): Lock
 - нет
- PauseAction.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку.

- Resume.setNewAction(Action a): void
 - a – новое действие
- ResumeAction.getLock(): Lock
 - нет
- ResumeAction.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку.

- TimerListener.actionPerformed(ActionEvent e): void
 - e – событие срабатывания таймера

- MainFrame.imagePanel(): BufferedImage
 - нет
- MainFrame.changeGraphicPanels(): void
 - нет

- ChooseGA.actionPerformed(ActionEvent e): void

- e – событие нажатия на кнопку
- ChooseIndividual.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку
- ExitAction.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку
- SaveGenerationAction.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку
- SavePictureAction.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку
- ShowGenerationAction.actionPerformed(ActionEvent e): void
 - e – событие нажатия на кнопку

- IndividualDialog.valueChanged(ListSelectionEvent e): void
 - e – событие изменения окна.
- IndividualDialog.actionPerformed(ActionEvent e): void
 - e – событие закрытия окна
- IndividualDialog.getSelectedIndex(): int
 - нет

- ShowTableModel.getColumnCount(): int
 - нет
- ShowTableModel.columnName(int index): String
 - index – номер колонки
- ShowTableModel.getRowCount(): int
 - нет
- ShowTableModel.isCellEditable(int row, int column): boolean
 - row – номер строки
 - column – номер столбца
- ShowTableModel.getColumnClass(int column): Class
 - column – номер столбца
- ShowTableModel.setValueAt(Object value, int row, int column): void
 - value – записываемое значение
 - row – номер строки
 - column – номер столбца
- ShowTableModel.getValueAt(int row, int column): Object - возвращает значение ячейки таблицы.
 - row – номер строки
 - column – номер столбца

- PluginLoader.getGA(int index, IndividualFactory factory): GA
 - index – номер алгоритма
 - factory – фабрика
- PluginLoader.getIndividualFactory(int index): IndividualFactory
 - index – номер фабрики
- PluginLoader.getName(int index, int number): String

- index – тип модуля
- number – номер модуля
- PluginLoader.getComment(int index, int number): String
 - index – тип модуля
 - number – номер модуля
- PluginLoader.getCount(int index): int
 - index – тип модуля
- PluginLoader.getIndividualMax(int index): double
 - index – номер особи
- PluginLoader.getIndividualCountSigns(int index): int
 - index – номер особи
- PluginLoader.getProperties(int index, int number): Properties
 - index – тип модуля
 - number – номер модуля
- PluginLoader.getFunctors(): Functor[]
 - нет
- PluginLoader.getFunctorTitle(int number): String
 - index – номер функтора

- GARunner.run(): void
 - нет
- GARunner.tryToInterrupt(): void
 - нет.
- GARunner.pause(): void
 - нет.
- GARunner.resume(): void
 - нет.
- GARunner.setLock(Lock lock): void
 - lock – устанавливаемая блокировка
- GARunner.getFunctors(): Functor[]
 - нет.
- GARunner.getIndividuals(int index): List<Individual>
 - index – номер поколения
- GARunner.getNumberGen(): List<Integer>
 - нет.
- GARunner.getCurrentGeneration(): int
 - нет.
- GARunner.setGA(GA ga): int
 - ga – устанавливаемый генетически алгоритм

- SelectedPlugin.setIndexGA(int i): void
 - i – номер алгоритма
- SelectedPlugin.getIndexGA(): int
 - нет
- SelectedPlugin.reDoGA(): void
 - нет
- SelectedPlugin.setIndividual(int i): void
 - i – номер представления
- SelectedPlugin.getIndexIndividual(): int

- нет
- SelectedPlugin.reDoIndividual(): void
 - нет

- RunnableLock.run(): void
 - нет

7. ВЫХОДНЫЕ ДАННЫЕ

- ChooseDialog.valueChanged(ListSelectionEvent e): void - нет;
 - ChooseDialog.actionPerformed(ActionEvent e): void – нет.
 - ConfigDialog.actionPerformed(ActionEvent e): void – нет.
-
- FrameCreator.run() :void – нет
-
- PauseAction.setNewAction(Action): void - нет
 - PauseAction.getLock(): Lock - возвращает блокировку.
 - PauseAction.actionPerformed(ActionEvent): void – нет.
-
- ResumeAction.setNewAction(Action): void - нет
 - ResumeAction.getLock(): Lock - возвращает блокировку.
 - ResumeAction.actionPerformed(ActionEvent): void – нет.
-
- TimerListener.actionPerformed(ActionEvent): void – нет
-
- MainFrame.imagePanel(): BufferedImage - возвращает изображение одной из вкладок.
 - MainFrame.changeGraphicPanels(): void - нет.
-
- ChooseGA.actionPerformed(ActionEvent e): void – нет
 - ExitAction.actionPerformed(ActionEvent e): void – нет
 - ChooseIndividual.actionPerformed(ActionEvent e): void – нет
 - SaveGenerationAction.actionPerformed(ActionEvent e): void – нет
 - SavePictureAction.actionPerformed(ActionEvent e): void – нет
-
- IndividualDialog.valueChanged(ListSelectionEvent): void - нет.
 - IndividualDialog.actionPerformed(ActionEvent): void - нет.
 - IndividualDialog.getSelectedIndex(): int - возвращает номер выделенной строки.
 - ShowTableModel.getColumnCount(): int - возвращает количество колонок.
 - ShowTableModel.columnName(int): String - возвращает название колонки по номеру.
 - ShowTableModel.getRowCount(): int - возвращает количество строчек.

- ShowTableModel.isCellEditable(int, int): boolean - возвращает true, если ячейку можно редактировать.
 - ShowTableModel.getColumnClass(int): Class - возвращает класс колонки по номеру.
 - ShowTableModel.setValueAt(Object, int, int): void - нет.
 - ShowTableModel.getValueAt(int, int): Object - возвращает значение ячейки таблицы.
-
- PluginLoader.getGA(int, IndividualFactory): GA - возвращает класс, работающий с популяцией.
 - PluginLoader.getIndividualFactory(int): IndividualFactory - возвращает класс указанного плагина, создающий произвольных особей.
 - PluginLoader.getName(int, int): String - возвращает название указанного плагина.
 - PluginLoader.getComment(int, int): String - возвращает комментарий к указанному плагину.
 - PluginLoader.getCount(int): int - возвращает количество указанных частей плагина.
 - PluginLoader.getIndividualMax(int): double - возвращает максимальное значение для графика.
 - PluginLoader.getIndividualCountSigns(int): int - возвращает количество отображаемых знаков фитнес-функции.
 - PluginLoader.getProperties(int, int): Properties - возвращает настройки указанной части плагина.
 - PluginLoader.getFunctors(): Functor[] - возвращает функцию графика.
 - PluginLoader.getFunctorTitle(int): String - возвращает название графика.
-
- GARunner.run(): void - нет
 - GARunner.tryToInterrupt(): void - нет.
 - GARunner.pause(): void - нет.
 - GARunner.resume(): void – нет.
 - GARunner.setLock(Lock lock): void - нет.
 - lock – устанавливаемая блокировка
 - GARunner.getFunctors(): Functor[] - возвращает графики функций.
 - GARunner.getIndividuals(int index): List<Individual> - возвращает список особей по индексу.
 - index – номер поколения
 - GARunner.getNumberGen(): List<Integer> - возвращает номера поколений, в которых появились лучшие особи.
 - GARunner.getCurrentGeneration(): int - возвращает номер текущего поколения.
 - GARunner.setGA(GA): int - устанавливает новый генетический алгоритм.
-
- SelectedPlugin.setIndexGA(int i): void - нет
 - SelectedPlugin.getIndexGA(): int - возвращает номер выбранного генетического алгоритма.
 - SelectedPlugin.reDoGA(): void - нет
 - SelectedPlugin.setIndividual(int i): void - нет
 - SelectedPlugin.getIndexIndividual(): int - возвращает номер представления особи.
 - SelectedPlugin.reDoIndividual(): void – нет

RunnableLock.run(): void – нет

