

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 GA.SIMPLE
ОПИСАНИЕ ПРОГРАММЫ

ЛИСТ УТВЕРЖДЕНИЯ

7.190.00001-01 13 05-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. зам. имя, И.	
Подп. и дата	

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 05-ЛУ

ПРОГРАММНОЕ СРЕДСТВО 3GENETIC

ПРОГРАММНЫЙ МОДУЛЬ GA.SIMPLE
ОПИСАНИЕ ПРОГРАММЫ

7.190.00001-01 13 05

Листов 13

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

АННОТАЦИЯ

В данном документе приводится описание модуля GA.simple программного средства 3GENETIC, реализующего классический вариант генетического алгоритма.

СОДЕРЖАНИЕ

Аннотация.....	2
Содержание.....	3
1. Общие сведения	4
2. Функциональное назначение	5
3. Описание логической структуры.....	6
3.1. Класс SimpleGA.....	6
3.1.1. Подробное описание.....	6
3.1.2. Защищенные члены	6
3.1.3. Открытые члены.....	6
3.1.4. Конструктор(ы)	6
3.1.5. Методы.....	7
3.2. Класс SimpleGALoader	7
3.2.1. Подробное описание.....	7
3.2.2. Защищенные члены	7
3.2.3. Открытые члены.....	7
3.2.4. Конструктор(ы)	7
3.2.5. Методы.....	8
4. Используемые технические средства.....	9
5. Вызов и загрузка	10
6. Входные данные.....	11
7. Выходные данные	12

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

Данный модуль является подключаемым (то есть программное средство 3Genetic способно работать в его отсутствии). Расширяет возможности ядра 3Genetic. Содержит реализацию классического генетического алгоритма.

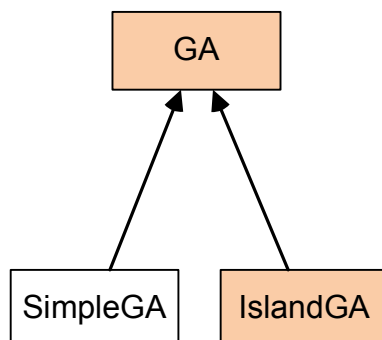
3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Класс SimpleGA

3.1.1. Подробное описание

Класс реализует классическую модель генетических алгоритмов.

Граф наследования:



3.1.2. Защищенные члены

- generation:List – популяция;
- sizeElite:int – число особей, напрямую переходящих в следующее поколение;
- probabilityMutation:double – вероятность мутации;
- factory:IndividualFactory – класс, производящий генерацию особей случайным образом;
- r:Random – генератор случайных чисел.

3.1.3. Открытые члены

- SimpleGA(int, double, double, IndividualFactory)
- nextGeneration():void
- getGeneration():List
- bigMutation():void
- getBest():Individual
- setGeneration(List):void

3.1.4. Конструктор(ы)

- SimpleGA(int sizeGeneration, double sizeElite, double probMuataion, IndividualFactory fact) – создает новый экземпляр генетического алгоритма.
 - sizeGeneration – размер поколения;
 - sizeElite – процент “элитных” особей;
 - probMutation - .вероятность метации особи;
 - IndividualFactory – фабрика случайных особей.
-

3.1.5. Методы

- `nextGeneration():void` – генерирует следующее поколение;
- `getGeneration():List` – возвращает текущее поколение;
- `bigMutation():void` – производит «большую» мутацию;
- `getBest():Individual` – возвращает наилучшую на данный момент особь;
- `setGeneration(List generation):void` – задает поколение.
 - `generation` – новое поколение

Объявления и описания членов класса находятся в файле:

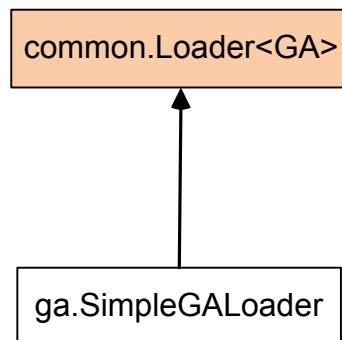
- `\src\ga\SimpleGA.java`

3.2. Класс SimpleGALoader

3.2.1. Подробное описание

Класс загружает классический генетический алгоритм

Граф наследования:



3.2.2. Защищенные члены

- `properties:Properties`

3.2.3. Открытые члены

- `SimpleGALoader(JarFile jr)`
- `load():GA`
- `getProperties():Properties`

3.2.4. Конструктор(ы)

- `SimpleGALoader(JarFile jr)` – создает новый загрузчик.

Аргументы:

- `jr` – разсер поколения.
-

3.2.5. Методы

- `load():GA` – генерирует новый экземпляр генетического алгоритма;
- `getProperties():Properties` – возвращает список возможных настроек алгоритма;

Объявления и описания членов класса находятся в файле:

- `\src\ga\SimpleGALoader.java`
 -
 -

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для нормального функционирования модуля Simple.ga 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 SimlpeGA(int, double, double, IndividualFactory)
public SimpleGA.getGeneration():List<Individual>
public SimpleGA.nextGeneration():void
public SimpleGA.bigMutation():void
public SimpleGA.setGenetaion():void
public SimpleGALoader(JarFile)
public SimpleGA SimpleGALoader.load():SimpleGA
public SimpleGALoader.getProperties():Properties
```

6. ВХОДНЫЕ ДАННЫЕ

- SimpleGA.SimpleGA(int sizeGeneration, double sizeElite, double probMutation, IndividualFactory fact)
 - sizeGeneration – размер поколения;
 - sizeElite – процент “элитных” особей;
 - probMutation - .вероятность метации особи;
 - IndividualFactory – фабрика случайных особей.
- SimpleGA.nextGeneration():void
 - нет
- SimpleGA.getGeneration():List
 - нет
- SimpleGA.bigMutation():void
 - нет
- SimpleGA.getBest():Individual
 - нет
- SimpleGA.setGeneration(List generation):void
 - generation – новое поколение
- SimpleGALoader.SimpleGALoader(JarFile jr)
 - jr – jar-архив в котором хранится данный модуль.
- SimpleGALoader.load():GA
 - нет
- SimpleGALoader.getProperties():Properties
 - нет

7. ВЫХОДНЫЕ ДАННЫЕ

- SimpleGA.SimpleGA(int sizeGeneration, double sizeElite, double probMuataion, IndividualFactory fact) – новый экземпляр генетического алгоритма
- SimpleGA.nextGeneration():void - нет
- SimpleGA.getGeneration():List - нет
- SimpleGA.bigMutation():void - нет
- SimpleGA.getBest():Individual – лучшая особь генетического алгоритма на текущий момент
- SimpleGA.setGeneration(List generation):void - нет
- SimpleGALoader.SimpleGALoader(JarFile jr) – новый экземпляр загрузчика
- SimpleGALoader.load():GA – новый экземпляр генетического алгоритма
- SimpleGALoader.getProperties():Properties – список настроек генетического алгоритма

