

# HarrixClass\_DataOfHarrixOptimizationTesting - HarrixClass\_DataOfHarrixOptimizationTesting v.1.25

А. Б. Сергиенко

17 июня 2014 г.

## Аннотация

Класс HarrixClass\_DataOfHarrixOptimizationTesting для считывания информации формата данных Harrix Optimization Testing на C++ для Qt. Рассматривается HarrixClass\_DataOfHarrixOptimizationTesting.cpp.

## Содержание

<b>1 Введение</b>	<b>5</b>
<b>2 Список функций</b>	<b>6</b>
<b>3 Функции</b>	<b>11</b>
3.1 _Конструкторы и деструкторы . . . . .	11
3.1.1 HarrixClass_DataOfHarrixOptimizationTesting . . . . .	11
3.1.2 ~HarrixClass_DataOfHarrixOptimizationTesting . . . . .	11
3.2 Возвращение HTML данных из класса . . . . .	11
3.2.1 getHtml . . . . .	11
3.2.2 getHtmlMessageOfError . . . . .	12
3.2.3 getHtmlReport . . . . .	12
3.3 Возвращение LaTeX данных из класса . . . . .	12
3.3.1 getFullLatex . . . . .	12
3.3.2 getFullLatexAboutParameters . . . . .	13
3.3.3 getFullLatexAnalysis . . . . .	13
3.3.4 getFullLatexInfo . . . . .	13
3.3.5 getFullLatexTable . . . . .	14

3.3.6	getFullLatexTableEx . . . . .	14
3.3.7	getFullLatexTableEy . . . . .	14
3.3.8	getFullLatexTableR . . . . .	15
3.3.9	getLatex . . . . .	15
3.3.10	getLatexAboutParameters . . . . .	15
3.3.11	getLatexAnalysis . . . . .	16
3.3.12	getLatexInfo . . . . .	16
3.3.13	getLatexTable . . . . .	16
3.3.14	getLatexTableEx . . . . .	17
3.3.15	getLatexTableEy . . . . .	17
3.3.16	getLatexTableR . . . . .	17
3.4	Возвращение переменных из класса . . . . .	18
3.4.1	getAuthor . . . . .	18
3.4.2	getCheckAllCombinations . . . . .	18
3.4.3	getData . . . . .	18
3.4.4	getDate . . . . .	18
3.4.5	getDimensionTestFunction . . . . .	19
3.4.6	getEmail . . . . .	19
3.4.7	getErrorEx . . . . .	19
3.4.8	getErrorEy . . . . .	20
3.4.9	getErrorR . . . . .	20
3.4.10	getFormat . . . . .	20
3.4.11	getFullNameAlgorithm . . . . .	21
3.4.12	getFullNameTestFunction . . . . .	21
3.4.13	getLink . . . . .	21
3.4.14	getLinkAlgorithm . . . . .	22
3.4.15	getLinkTestFunction . . . . .	22
3.4.16	getMaxCountOfFitness . . . . .	22
3.4.17	getMeanEx . . . . .	22
3.4.18	getMeanEy . . . . .	23
3.4.19	getMeanR . . . . .	23
3.4.20	getNameAlgorithm . . . . .	23

3.4.21	getNameParameter . . . . .	24
3.4.22	getNameTestFunction . . . . .	24
3.4.23	getNumberOfExperiments . . . . .	24
3.4.24	getNumberOfMeasuring . . . . .	25
3.4.25	getNumberOfOption . . . . .	25
3.4.26	getNumberOfParameters . . . . .	25
3.4.27	getNumberOfRuns . . . . .	26
3.4.28	getParameter . . . . .	26
3.4.29	getSuccessReading . . . . .	26
3.4.30	getVarianceOfEx . . . . .	27
3.4.31	getVarianceOfEy . . . . .	27
3.4.32	getVarianceOfR . . . . .	27
3.4.33	getVersion . . . . .	27
3.5	Операторы . . . . .	28
3.5.1	operator = . . . . .	28
3.6	Специализированные функции . . . . .	28
3.6.1	makingLatexTable2D . . . . .	28
3.7	Внутренние функции . . . . .	29
3.7.1	checkXmlLeafTags . . . . .	29
3.7.2	getLatexBegin . . . . .	29
3.7.3	getLatexEnd . . . . .	29
3.7.4	initializationOfVariables . . . . .	29
3.7.5	makingListOfVectorParameterOptions . . . . .	30
3.7.6	memoryAllocation . . . . .	30
3.7.7	memoryDeallocation . . . . .	30
3.7.8	readXml . . . . .	31
3.7.9	readXmlDataTags . . . . .	31
3.7.10	readXmlLeafTag . . . . .	31
3.7.11	readXmlTreeTag . . . . .	32
3.7.12	zeroArray . . . . .	32
3.8	Создание содержимого отчетов LaTeX и HTML . . . . .	32
3.8.1	makingHtmlReport . . . . .	32

3.8.2	<code>makingLatexAboutParameters</code> . . . . .	33
3.8.3	<code>makingLatexAnalysis</code> . . . . .	33
3.8.4	<code>makingLatexInfo</code> . . . . .	33
3.8.5	<code>makingLatexListOfVectorParameterOptions</code> . . . . .	34
3.8.6	<code>makingLatexTableEx</code> . . . . .	34
3.8.7	<code>makingLatexTableEy</code> . . . . .	34
3.8.8	<code>makingLatexTableR</code> . . . . .	35
3.9	Функции анализа данных . . . . .	35
3.9.1	<code>makingAnalysis</code> . . . . .	35

# 1 Введение

Класс `HarrixClass_DataOfHarrixOptimizationTesting` для считывания информации формата данных Harrix Optimization Testing на C++ для Qt.

Последнюю версию документа можно найти по адресу:

[https://github.com/Harrix/HarrixClass\\_DataOfHarrixOptimizationTesting](https://github.com/Harrix/HarrixClass_DataOfHarrixOptimizationTesting)

Об установке библиотеки можно прочитать тут:

<http://blog.harrix.org/?p=992>

С автором можно связаться по адресу [sergienkoanton@mail.ru](mailto:sergienkoanton@mail.ru) или <http://vk.com/harrix>.

Сайт автора, где публикуются последние новости: <http://blog.harrix.org/>, а проекты располагаются по адресу <http://harrix.org/>.

## 2 Список функций

### Конструкторы и деструкторы

1. **HarrixClass\_DataOfHarrixOptimizationTesting** — Конструктор. Функция считывает данные о тестировании алгоритма оптимизации из файла формата HarrixOptimizationTesting. Во второй реализации это конструктор., который создает пустой экземпляр.
2. **~HarrixClass\_DataOfHarrixOptimizationTesting** — Деструктор класса.

### **Возвращение HTML данных из класса**

1. **getHtml** — Получение текста переменной Html. Это итоговый Html документ. Помните, что это не полноценный Html код. Его нужно применять в виде temp.html для макета: <https://github.com/Harrix/QtHarrixLibraryForQWebView>.
2. **getHtmlMessageOfError** — Получение текста переменной HtmlMessageOfError. Это часть html документа в виде кода о сообщениях ошибок. Помните, что это не полноценный Html код. Его нужно применять в виде temp.html для макета: <https://github.com/Harrix/QtHarrixLibraryForQWebView>.
3. **getHtmlReport** — Получение текста переменной HtmlReport. Это часть html документа в виде отчета о проделанной работе. Помните, что это не полноценный Html код. Его нужно применять в виде temp.html для макета: <https://github.com/Harrix/QtHarrixLibraryForQWebView>.

### **Возвращение LaTeX данных из класса**

1. **getFullLatex** — Получение текста переменной Latex в полном составе с началом и концовкой в Latex файле. Здесь собран полный файл анализа данных из считываемого xml файла. Это полноценный Latex код. Его нужно применять с файлами из макета: <https://github.com/Harrix/Harrix-Document-Template-LaTeX>.
2. **getFullLatexAboutParameters** — Получение текста переменной LatexAboutParameters — отображение данных о обнаруженных параметрах алгоритма и какие они бывают с началом и концовкой в Latex файле.
3. **getFullLatexAnalysis** — Получение текста переменной LatexAnalysis — отображение данных первоначального анализа данных.
4. **getFullLatexInfo** — Получение текста переменной LatexInfo — отображение информации о исследовании с началом и концовкой в Latex файле.
5. **getFullLatexTable** — Получение текста переменной LatexTable в полном составе с началом и концовкой в Latex файле.
6. **getFullLatexTableEx** — Получение текста переменной LatexTableEx — отображение сырых данных таблицы данных о ошибке Ex с началом и концовкой в Latex файле.
7. **getFullLatexTableEy** — Получение текста переменной LatexTableEy — отображение сырых данных таблицы данных о ошибке Ey с началом и концовкой в Latex файле.
8. **getFullLatexTableR** — Получение текста переменной LatexTableR — отображение сырых данных по надежности в виде полной таблицы с началом и концовкой в Latex файле.

9. **getLatex** — Получение текста переменной Latex. Здесь собран полный файл анализа данных из считываемого xml файла. Помните, что это не полноценный Latex код. Его нужно применять внутри файла из макета: <https://github.com/Harrix/Harrix-Document-Template-LaTeX>.
10. **getLatexAboutParameters** — Получение текста переменной LatexAboutParameters — отображение данных о обнаруженных параметрах алгоритма и какие они бывают.
11. **getLatexAnalysis** — Получение текста переменной LatexAnalysis — отображение первоначального анализа данных.
12. **getLatexInfo** — Получение текста переменной LatexInfo — отображение информации о исследовании.
13. **getLatexTable** — Получение текста переменной LatexTable.
14. **getLatexTableEx** — Получение текста переменной LatexTableEx — отображение сырых данных таблицы данных о ошибке Ex.
15. **getLatexTableEy** — Получение текста переменной LatexTableEy — отображение сырых данных ошибки по значениям целевой функции в виде полной таблицы.
16. **getLatexTableR** — Получение текста переменной LatexTableR — отображение сырых данных по надежности в виде полной таблицы.

### Возвращение переменных из класса

1. **getAuthor** — Получение текста переменной XML\_Author — Автор документа.
2. **getCheckAllCombinations** — Получение текста переменной XML\_All\_Combinations — Все ли комбинации вариантов настроек просмотрены: 0 или 1.
3. **getData** — Возвращает содержимое исследований в виде экземпляра класса.
4. **getDate** — Получение текста переменной XML\_Date — Дата создания документа.
5. **getDimensionTestFunction** — Получение текста переменной XML\_DimensionTestFunction — Размерность тестовой задачи.
6. **getEmail** — Получение текста переменной XML\_Email — Email автора, чтобы можно было с ним связаться
7. **getErrorEx** — Получение значения ошибки Ex.
8. **getErrorEy** — Получение значения ошибки Ey.
9. **getErrorR** — Получение значения надежности R.
10. **getFormat** — Получение переменной XML\_Format, то есть возвращает название формата документа.
11. **getFullNameAlgorithm** — Получение текста переменной XML\_Full\_Name\_Algorithm — Полное название алгоритма оптимизации.
12. **getFullNameTestFunction** — Получение текста переменной XML\_Full\_Name\_Test\_Function — Полное название тестовой функции.

13. **getLink** — Получение переменной XML\_Link, то есть возвращает ссылку на описание формата файла.
14. **getLinkAlgorithm** — Получение текста переменной XML\_Link\_Algorithm — Ссылка на описание алгоритма оптимизации.
15. **getLinkTestFunction** — Получение текста переменной XML\_Link\_Test\_Function — Ссылка на описание тестовой функции.
16. **getMaxCountOfFitness** — Получение текста переменной Max\_Count\_Of\_Fitness — Максимальное допустимое число вычислений целевой функции для алгоритма.
17. **getMeanEx** — Получение среднего значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).
18. **getMeanEy** — Получение среднего значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).
19. **getMeanR** — Получение среднего значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).
20. **getNameAlgorithm** — Получение текста переменной XML\_Name\_Algorithm — Название алгоритма оптимизации.
21. **getNameParameter** — Получение значения параметра настройки какой-то в виде полного наименования.
22. **getNameTestFunction** — Получение текста переменной XML\_Name\_Test\_Function — Название тестовой функции.
23. **getNumberOfExperiments** — Получение текста переменной XML\_Number\_Of\_Experiments — Количество комбинаций вариантов настроек.
24. **getNumberOfMeasuring** — Получение текста переменной XML\_Number\_Of\_Measuring — Размерность тестовой задачи (длина хромосомы решения).
25. **getNumberOfOption** — Получение номера параметра алгоритма по его имени.
26. **getNumberOfParameters** — Получение текста переменной XML\_Number\_Of\_Parameters — Количество проверяемых параметров алгоритма оптимизации
27. **getNumberOfRuns** — Получение текста переменной XML\_Number\_Of\_Runs — Количество прогонов, по которому делается усреднение для эксперимента.
28. **getParameter** — Получение значения параметра настройки какой-то.
29. **getSuccessReading** — Получение значения переменной SuccessReading о удачности или не удачности считывания файла.
30. **getVarianceOfEx** — Получение дисперсии значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).
31. **getVarianceOfEy** — Получение дисперсии значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).



32. **getVarianceOfR** — Получение дисперсии значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).
33. **getVersion** — Получение переменной XML\_Version, то есть возвращает версию формата документа.

## Операторы

1. **operator =** — Оператор присваивания.

## Специализированные функции

1. **makingLatexTable2D** — Создает текст LaTeX в виде таблицы 2D для всех экспериментов для отображения какой-нибудь информации.

## Внутренние функции

1. **checkXmlLeafTags** — Проверяет наличие тэгов и правильное их выполнение. Внутренняя функция. Учитывает все "листовые" тэги кроме тэгов данных.
2. **getLatexBegin** — Внутренняя функция. Возвращает начало для полноценного Latex файла.
3. **getLatexEnd** — Внутренняя функция. Возвращает концовку для полноценного Latex файла.
4. **initializationOfVariables** — Обнуление переменных. Внутренняя функция.
5. **makingListOfVectorParameterOptions** — Заполняет список вектора названий вариантов параметров алгоритма оптимизации.
6. **memoryAllocation** — Выделяет память под необходимые массивы. Внутренняя функция.
7. **memoryDeallocation** — Удаляет память из-под массивов. Внутренняя функция.
8. **readXml** — Считывание XML файла и осуществление всех остальных анализов и др.
9. **readXmlDataTags** — Считывает и проверяет тэги данных. Внутренняя функция. Учитывает все "листовые" тэги кроме тэгов данных.
10. **readXmlLeafTag** — Считывает и проверяет тэг, который должен являться "листом то есть самым глубоким. Внутренняя функция. Учитывает все "листовые" тэги кроме тэгов данных.
11. **readXmlTreeTag** — Считывает и проверяет тэг, который содержит внутри себя другие тэги. Внутренняя функция.
12. **zeroArray** — Обнуляет массивы, в которые записывается информация о данных из файла. Внутренняя функция.

## Создание содержимого отчетов LaTeX и HTML

1. **makingHtmlReport** — Создает текст Html для отображения отчета о считывании XML файла.
2. **makingLatexAboutParameters** — Создает текст LaTeX для отображения данных о обнаруженных параметрах алгоритма и какие они бывают.

3. **makingLatexAnalysis** — Создает текст LaTeX для отображения первоначального анализа данных.
4. **makingLatexInfo** — Создает текст LaTeX для отображения информации о исследовании.
5. **makingLatexListOfVectorParameterOptions** — Создает текст LaTeX для отображения списка вектора названий вариантов параметров алгоритма оптимизации.
6. **makingLatexTableEx** — Создает текст LaTeX для отображения сырых данных ошибки по входным параметрам в виде полной таблицы.
7. **makingLatexTableEy** — Создает текст LaTeX для отображения сырых данных ошибки по значениям целевой функции в виде полной таблицы.
8. **makingLatexTableR** — Создает текст LaTeX для отображения сырых данных по надежности в виде полной таблицы.

#### **Функции анализа данных**

1. **makingAnalysis** — Выполняет анализ считанных данных. Внутренняя функция.

## 3 Функции

### 3.1 \_Конструкторы и деструкторы

#### 3.1.1 HarrixClass\_DataOfHarrixOptimizationTesting

Конструктор. Функция считывает данные о тестировании алгоритма оптимизации из файла формата HarrixOptimizationTesting. Во второй реализации это конструктор., который создает пустой экземпляр.

##### Код 1. Синтаксис

```
HarrixClass_DataOfHarrixOptimizationTesting(QString filename);  
HarrixClass_DataOfHarrixOptimizationTesting();
```

##### Входные параметры:

filename — полное имя считываемого файла.

##### Возвращаемое значение:

Отсутствует.

#### 3.1.2 ~HarrixClass\_DataOfHarrixOptimizationTesting

Деструктор класса.

##### Код 2. Синтаксис

```
~HarrixClass_DataOfHarrixOptimizationTesting();
```

##### Входные параметры:

Отсутствуют.

##### Возвращаемое значение:

Отсутствует.

### 3.2 Возвращение HTML данных из класса

#### 3.2.1 getHtml

Получение текста переменной Html. Это итоговый Html документ. Помните, что это не полноценный Html код. Его нужно применять в виде temp.html для макета: <https://github.com/Harrix/QtHarrixLibraryForQWebView>.

##### Код 3. Синтаксис

```
QString getHtml();
```

##### Входные параметры:

Отсутствуют.

**Возвращаемое значение:**

Итоговый Html документ.

### 3.2.2 getHtmlMessageOfError

Получение текста переменной HtmlMessageOfError. Это часть html документа в виде кода о сообщениях ошибок. Помните, что это не полноценный Html код. Его нужно применять в виде temp.html для макета: <https://github.com/Harrix/QtHarrixLibraryForQWebView>.

Код 4. Синтаксис

```
QString getHtmlMessageOfError();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Часть html документа в виде кода о сообщениях ошибок.

### 3.2.3 getHtmlReport

Получение текста переменной HtmlReport. Это часть html документа в виде отчета о проделанной работе. Помните, что это не полноценный Html код. Его нужно применять в виде temp.html для макета: <https://github.com/Harrix/QtHarrixLibraryForQWebView>.

Код 5. Синтаксис

```
QString getHtmlReport();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Часть html документа в виде отчета о проделанной работе.

## 3.3 Возвращение LaTeX данных из класса

### 3.3.1 getFullLatex

Получение текста переменной Latex в полном составе с началом и концовкой в Latex файле. Здесь собран полный файл анализа данных из считываемого xml файла. Это полноценный Latex код. Его нужно применять с файлами из макета: <https://github.com/Harrix/Harrix-Document-Template-Latex>.

Код 6. Синтаксис

```
QString getFullLatex();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Полный файл анализа данных из считываемого xml файла.

### 3.3.2 getFullLatexAboutParameters

Получение текста переменной `LatexAboutParameters` — отображение данных о обнаруженных параметрах алгоритма и какие они бывают с началом и концовкой в Latex файле.

Код 7. Синтаксис

```
QString getFullLatexAboutParameters();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Полный переменной `LatexAboutParameters` — отображение данных о обнаруженных параметрах алгоритма и какие они бывают с началом и концовкой в Latex файле.

### 3.3.3 getFullLatexAnalysis

Получение текста переменной `LatexAnalysis` — отображение данных первоначального анализа данных.

Код 8. Синтаксис

```
QString getFullLatexAnalysis();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной `LatexAnalysis` — отображение данных первоначального анализа данных.

### 3.3.4 getFullLatexInfo

Получение текста переменной `LatexInfo` — отображение информации о исследовании с началом и концовкой в Latex файле.

Код 9. Синтаксис

```
QString getFullLatexInfo();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной `LatexInfo` — отображение информации о исследовании с началом и концовкой в `Latex` файле.

**3.3.5 `getFullLatexTable`**

Получение текста переменной `LatexTable` в полном составе с началом и концовкой в `Latex` файле.

Код 10. Синтаксис

```
QString getFullLatexTable();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Полный файл первичных данных из считываемого `xml` файла (без анализа).

**3.3.6 `getFullLatexTableEx`**

Получение текста переменной `LatexTableEx` — отображение сырых данных таблицы данных о ошибке `Ex` с началом и концовкой в `Latex` файле.

Код 11. Синтаксис

```
QString getFullLatexTableEx();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной `LatexTableEx` — отображение сырых данных таблицы данных о ошибке `Ex` с началом и концовкой в `Latex` файле.

**3.3.7 `getFullLatexTableEy`**

Получение текста переменной `LatexTableEy` — отображение сырых данных таблицы данных о ошибке `Ey` с началом и концовкой в `Latex` файле.

Код 12. Синтаксис

```
QString getFullLatexTableEy();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной `LatexTableEy` — отображение сырых данных таблицы данных о ошибке `Ey` с началом и концовкой в `Latex` файле.

### 3.3.8 `getFullLatexTableR`

Получение текста переменной `LatexTableR` — отображение сырых данных по надежности в виде полной таблицы с началом и концовкой в `Latex` файле.

Код 13. Синтаксис

```
QString getFullLatexTableR();
```

#### **Входные параметры:**

Отсутствуют.

#### **Возвращаемое значение:**

Текст переменной `LatexTableR` — отображение сырых данных по надежности в виде полной таблицы с началом и концовкой в `Latex` файле.

### 3.3.9 `getLatex`

Получение текста переменной `Latex`. Здесь собран полный файл анализа данных из считываемого `xml` файла. Помните, что это не полноценный `Latex` код. Его нужно применять внутри файла из макета: <https://github.com/Harrix/Harrix-Document-Template-LaTeX>.

Код 14. Синтаксис

```
QString getLatex();
```

#### **Входные параметры:**

Отсутствуют.

#### **Возвращаемое значение:**

Полный файл анализа данных из считываемого `xml` файла.

### 3.3.10 `getLatexAboutParameters`

Получение текста переменной `LatexAboutParameters` — отображение данных о обнаруженных параметрах алгоритма и какие они бывают.

Код 15. Синтаксис

```
QString getLatexAboutParameters();
```

#### **Входные параметры:**

Отсутствуют.

#### **Возвращаемое значение:**

Текст переменной `LatexAboutParameters` — отображение данных о обнаруженных параметрах алгоритма и какие они бывают.

### 3.3.11 `getLatexAnalysis`

Получение текста переменной `LatexAnalysis` — отображение первоначального анализа данных.

Код 16. Синтаксис

```
QString getLatexAnalysis();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Полный переменной `LatexAnalysis` — отображение первоначального анализа данных.

### 3.3.12 `getLatexInfo`

Получение текста переменной `LatexInfo` — отображение информации о исследовании.

Код 17. Синтаксис

```
QString getLatexInfo();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной `LatexInfo` — отображение информации о исследовании.

### 3.3.13 `getLatexTable`

Получение текста переменной `LatexTable`.

Код 18. Синтаксис

```
QString getLatexTable();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Полный файл первичных данных из считываемого xml файла (без анализа).



### 3.3.14 getLatexTableEx

Получение текста переменной LatexTableEx — отображение сырых данных таблицы данных о ошибке Ex.

Код 19. Синтаксис

```
QString getLatexTableEx();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной LatexTableEx — отображение сырых данных таблицы данных о ошибке Ex.

### 3.3.15 getLatexTableEy

Получение текста переменной LatexTableEy — отображение сырых данных ошибки по значениям целевой функции в виде полной таблицы.

Код 20. Синтаксис

```
QString getLatexTableEy();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной LatexTableEy — отображение сырых данных таблицы данных о ошибке Ey.

### 3.3.16 getLatexTableR

Получение текста переменной LatexTableR — отображение сырых данных по надежности в виде полной таблицы.

Код 21. Синтаксис

```
QString getLatexTableR();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Текст переменной LatexTableR — отображение сырых данных по надежности в виде полной таблицы.

## 3.4 Возвращение переменных из класса

### 3.4.1 getAuthor

Получение текста переменной XML\_Author — Автор документа.

Код 22. Синтаксис

```
QString getAuthor();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.2 getCheckAllCombinations

Получение текста переменной XML\_All\_Combinations — Все ли комбинации вариантов настроек просмотрены: 0 или 1.

Код 23. Синтаксис

```
bool getCheckAllCombinations();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.3 getData

Возвращает содержимое исследований в виде экземпляра класса.

Код 24. Синтаксис

```
HarrixClass_OnlyDataOfHarrixOptimizationTesting& getData();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Содержимое исследований в виде экземпляра класса.

### 3.4.4 getDate

Получение текста переменной XML\_Date — Дата создания документа.

Код 25. Синтаксис

```
QString getDate();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.5 getDimensionTestFunction

Получение текста переменной XML\_DimensionTestFunction — Размерность тестовой задачи.

Код 26. Синтаксис

```
qint64 getDimensionTestFunction();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.6 getEmail

Получение текста переменной XML\_Email — Email автора, чтобы можно было с ним связаться

Код 27. Синтаксис

```
QString getEmail();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.7 getErrorEx

Получение значения ошибки Ex.

Код 28. Синтаксис

```
double getErrorEx(int Number_Of_Experiment, int Number_Of_Measuring);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Measuring — номер измерения варианта настроек.

**Возвращаемое значение:**

Значения ошибки Ex.

### 3.4.8 getErrorEy

Получение значения ошибки Ey.

Код 29. Синтаксис

```
double getErrorEy(int Number_Of_Experiment, int Number_Of_Measuring);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Measuring — номер измерения варианта настроек.

**Возвращаемое значение:**

Значения ошибки Ey.

### 3.4.9 getErrorR

Получение значения надежности R.

Код 30. Синтаксис

```
double getErrorR(int Number_Of_Experiment, int Number_Of_Measuring);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Measuring — номер измерения варианта настроек.

**Возвращаемое значение:**

Значения надежности R.

### 3.4.10 getFormat

Получение переменной XML\_Format, то есть возвращает название формата документа.

Код 31. Синтаксис

```
QString getFormat();
```

**Входные параметры:**

Отсутствует.

**Возвращаемое значение:**

Если документ без ошибок в описании формата, то всегда должно возвращаться "Harrix Optimization Testing".

**3.4.11 getFullNameAlgorithm**

Получение текста переменной XML\_Full\_Name\_Algorithm — Полное название алгоритма оптимизации.

Код 32. Синтаксис

```
QString getFullNameAlgorithm();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

**3.4.12 getFullNameTestFunction**

Получение текста переменной XML\_Full\_Name\_Test\_Function — Полное название тестовой функции.

Код 33. Синтаксис

```
QString getFullNameTestFunction();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

**3.4.13 getLink**

Получение переменной XML\_Link, то есть возвращает ссылку на описание формата файла.

Код 34. Синтаксис

```
QString getLink();
```

**Входные параметры:**

Отсутствует.

**Возвращаемое значение:**

Если документ без ошибок в описании формата, то всегда должно возвращаться "https://github.com/Harrix/HarrixFileFormats".

### 3.4.14 getLinkAlgorithm

Получение текста переменной XML\_Link\_Algorithm — Ссылка на описание алгоритма оптимизации.

Код 35. Синтаксис

```
QString getLinkAlgorithm();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.15 getLinkTestFunction

Получение текста переменной XML\_Link\_Test\_Function — Ссылка на описание тестовой функции.

Код 36. Синтаксис

```
QString getLinkTestFunction();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.16 getMaxCountOfFitness

Получение текста переменной Max\_Count\_Of\_Fitness — Максимальное допустимое число вычислений целевой функции для алгоритма.

Код 37. Синтаксис

```
qint64 getMaxCountOfFitness();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.17 getMeanEx

Получение среднего значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).

Код 38. Синтаксис

```
double getMeanEx(int Number_Of_Experiment);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения среднего значения Ex.

### 3.4.18 getMeanEy

Получение среднего значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).

Код 39. Синтаксис

```
double getMeanEy(int Number_Of_Experiment);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения среднего значения Ey.

### 3.4.19 getMeanR

Получение среднего значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).

Код 40. Синтаксис

```
double getMeanR(int Number_Of_Experiment);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения среднего значения R.

### 3.4.20 getNameAlgorithm

Получение текста переменной XML\_Name\_Algorithm — Название алгоритма оптимизации.

Код 41. Синтаксис

```
QString getNameAlgorithm();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.21 getNameParameter

Получение значения параметра настройки какой-то в виде полного наименования.

Код 42. Синтаксис

```
QString getNameParameter(int Number_Of_Experiment, int Number_Of_Parameter);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Parameter — номер параметра.

**Возвращаемое значение:**

Значения параметра в виде наименования.

### 3.4.22 getNameTestFunction

Получение текста переменной XML\_Name\_Test\_Function — Название тестовой функции.

Код 43. Синтаксис

```
QString getNameTestFunction();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.23 getNumberOfExperiments

Получение текста переменной XML\_Number\_Of\_Experiments — Количество комбинаций вариантов настроек.

Код 44. Синтаксис

```
qint64 getNumberOfExperiments();
```

**Входные параметры:**

Отсутствуют.



**Возвращаемое значение:**

Значение переменной из описания.

**3.4.24 getNumberOfMeasuring**

Получение текста переменной XML\_Number\_Of\_Measuring — Размерность тестовой задачи (длина хромосомы решения).

Код 45. Синтаксис

```
qint64 getNumberOfMeasuring();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

**3.4.25 getNumberOfOption**

Получение номера параметра алгоритма по его имени.

Код 46. Синтаксис

```
int getNumberOfOption(QString NameOption);
```

**Входные параметры:**

NameOption — имя параметра.

**Возвращаемое значение:**

Значения параметра в виде номера (если не найдено, то возвращается -1).

**3.4.26 getNumberOfParameters**

Получение текста переменной XML\_Number\_Of\_Parameters — Количество проверяемых параметров алгоритма оптимизации

Код 47. Синтаксис

```
qint64 getNumberOfParameters();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.27 getNumberOfRuns

Получение текста переменной XML\_Number\_Of\_Runs — Количество прогонов, по которому делается усреднение для эксперимента.

Код 48. Синтаксис

```
qint64 getNumberOfRuns();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.4.28 getParameter

Получение значения параметра настройки какой-то.

Код 49. Синтаксис

```
int getParameter(int Number_Of_Experiment, int Number_Of_Parameter);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Parameter — номер параметра.

**Возвращаемое значение:**

Значения параметра в виде числа (соответствие находим в ListOfParameterOptions).

### 3.4.29 getSuccessReading

Получение значения переменной SuccessReading о удачности или не удачности считывания файла.

Код 50. Синтаксис

```
bool getSuccessReading();
```

**Входные параметры:**

Отсутствует.

**Возвращаемое значение:**

Значения переменной SuccessReading о удачности или не удачности считывания файла..

### 3.4.30 getVarianceOfEx

Получение дисперсии значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).

Код 51. Синтаксис

```
double getVarianceOfEx(int Number_Of_Experiment);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения дисперсии значения Ex.

### 3.4.31 getVarianceOfEy

Получение дисперсии значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).

Код 52. Синтаксис

```
double getVarianceOfEy(int Number_Of_Experiment);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения дисперсии значения Ey.

### 3.4.32 getVarianceOfR

Получение дисперсии значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).

Код 53. Синтаксис

```
double getVarianceOfR(int Number_Of_Experiment);
```

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения дисперсии значения надежности R.

### 3.4.33 getVersion

Получение переменной XML\_Version, то есть возвращает версию формата документа.

Код 54. Синтаксис

```
QString getVersion();
```

**Входные параметры:**

Отсутствует.

**Возвращаемое значение:**

Если документ без ошибок в описании формата, то всегда должно возвращаться "1.0".

## 3.5 Операторы

### 3.5.1 operator =

Оператор присваивания.

Код 55. Синтаксис

```
void operator = (HarrixClass_DataOfHarrixOptimizationTesting& B);
```

**Входные параметры:**

B — Другой экземпляр класса, который и копируем.

**Возвращаемое значение:**

Отсутствует.

## 3.6 Специализированные функции

### 3.6.1 makingLatexTable2D

Создает текст LaTeX в виде таблицы 2D для всех экспериментов для отображения какой-нибудь информации.

Код 56. Синтаксис

```
QString makingLatexTable2D(QString Title, QStringList InfoForEveryExperiment);
```

**Входные параметры:**

Title — заголовок таблицы;

InfoForEvryExperiment — информация выдаваемая в таблицу.

**Возвращаемое значение:**

Итоговая таблица в виде кода Latex.

## 3.7 Внутренние функции

### 3.7.1 checkXmlLeafTags

Проверяет наличие тэгов и правильное их выполнение. Внутренняя функция. Учитывает все "листовые" тэги кроме тэгов данных.

Код 57. Синтаксис

```
void checkXmlLeafTags();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.2 getLatexBegin

Внутренняя функция. Возвращает начало для полноценного Latex файла.

Код 58. Синтаксис

```
QString getLatexBegin();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Начало для полноценного Latex файла.

### 3.7.3 getLatexEnd

Внутренняя функция. Возвращает концовку для полноценного Latex файла.

Код 59. Синтаксис

```
QString getLatexEnd();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Концовка для полноценного Latex файла.

### 3.7.4 initializationOfVariables

Обнуление переменных. Внутренняя функция.

Код 60. Синтаксис

```
void initializationOfVariables();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.5 makingListOfVectorParameterOptions

Заполняет список вектора названий вариантов параметров алгоритма оптимизации.

Код 61. Синтаксис

```
void makingListOfVectorParameterOptions();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexListOfParameterOptions.

### 3.7.6 memoryAllocation

Выделяет память под необходимые массивы. Внутренняя функция.

Код 62. Синтаксис

```
void memoryAllocation();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.7 memoryDeallocation

Удаляет память из-под массивов. Внутренняя функция.

Код 63. Синтаксис

```
void memoryDeallocation();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.8 readXml

Считывание XML файла и осуществление всех остальных анализов и др.

Код 64. Синтаксис

```
void readXml();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.9 readXmlDataTags

Считывает и проверяет тэги данных. Внутренняя функция. Учитывает все "листовые" тэги кроме тэгов данных.

Код 65. Синтаксис

```
void readXmlDataTags();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.10 readXmlLeafTag

Считывает и проверяет тэг, который должен являться "листом то есть самым глубоким. Внутренняя функция. Учитывает все "листовые" тэги кроме тэгов данных.

Код 66. Синтаксис

```
void readXmlLeafTag();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.7.11 readXmlTreeTag

Считывает и проверяет тэг, который содержит внутри себя другие тэги. Внутренняя функция.

Код 67. Синтаксис

```
bool readXmlTreeTag(QString tag);
```

#### Входные параметры:

tag — какой тэг мы ищем.

#### Возвращаемое значение:

true — текущий тэг это тот самый, что нам и нужен;

false — иначе.

### 3.7.12 zeroArray

Обнуляет массивы, в котрые записывается информация о данных из файла. Внутренняя функция.

Код 68. Синтаксис

```
void zeroArray();
```

#### Входные параметры:

Отсутствуют.

#### Возвращаемое значение:

Отсутствует.

## 3.8 Создание содержимого отчетов LaTeX и HTML

### 3.8.1 makingHtmlReport

Создает текст Html для отображения отчета о считывании XML файла.

Код 69. Синтаксис

```
void makingHtmlReport();
```

#### Входные параметры:

Отсутствуют.

#### Возвращаемое значение:

Отсутствует. Значение возвращается в переменную HtmlReport, которую можно вызвать getHtmlReport.



### 3.8.2 makingLatexAboutParameters

Создает текст LaTeX для отображения данных о обнаруженных параметрах алгоритма и какие они бывают.

Код 70. Синтаксис

```
void makingLatexAboutParameters();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexTableEx, которую можно вызвать getLatexAboutParameters

### 3.8.3 makingLatexAnalysis

Создает текст LaTeX для отображения первоначального анализа данных.

Код 71. Синтаксис

```
void makingLatexAnalysis();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexAnalysis, которую можно вызвать getLatexAnalysis.

### 3.8.4 makingLatexInfo

Создает текст LaTeX для отображения информации о исследовании.

Код 72. Синтаксис

```
void makingLatexInfo();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexTableEx, которую можно вызвать getLatexInfo.

### 3.8.5 makingLatexListOfVectorParameterOptions

Создает текст LaTeX для отображения списка вектора названий вариантов параметров алгоритма оптимизации.

Код 73. Синтаксис

```
void makingLatexListOfVectorParameterOptions();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexListOfParameterOptions.

### 3.8.6 makingLatexTableEx

Создает текст LaTeX для отображения сырых данных ошибки по входным параметрам в виде полной таблицы.

Код 74. Синтаксис

```
void makingLatexTableEx();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexTableEx, которую можно вызвать getLatexTableEx.

### 3.8.7 makingLatexTableEy

Создает текст LaTeX для отображения сырых данных ошибки по значениям целевой функции в виде полной таблицы.

Код 75. Синтаксис

```
void makingLatexTableEy();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexTableEy, которую можно вызвать getLatexTableEy.

### 3.8.8 makingLatexTableR

Создает текст LaTeX для отображения сырых данных по надежности в виде полной таблицы.

Код 76. Синтаксис

```
void makingLatexTableR();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует. Значение возвращается в переменную LatexTableR, которую можно вызвать getLatexTableR.

## 3.9 Функции анализа данных

### 3.9.1 makingAnalysis

Выполняет анализ считанных данных. Внутренняя функция.

Код 77. Синтаксис

```
void makingAnalysis();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.