

# HarrixClass\_OnlyDataOfHarrixOptimizationTesting - HarrixClass\_DataOfHarrixOptimizationTesting v.1.27

А. Б. Сергиенко

25 февраля 2015 г.

## Аннотация

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

## Оглавление

<b>1 Введение</b>	<b>5</b>
<b>2 Список функций</b>	<b>6</b>
<b>3 Функции</b>	<b>12</b>
3.1 _Конструкторы и деструкторы . . . . .	12
3.1.1 HarrixClass_OnlyDataOfHarrixOptimizationTesting . . . . .	12
3.1.2 ~HarrixClass_OnlyDataOfHarrixOptimizationTesting . . . . .	12
3.2 Возвращение данных из класса . . . . .	12
3.2.1 getAuthor . . . . .	12
3.2.2 getCheckAllCombinations . . . . .	13
3.2.3 getDate . . . . .	13
3.2.4 getDimensionTestFunction . . . . .	13
3.2.5 getEmail . . . . .	14
3.2.6 getErrorEx . . . . .	14
3.2.7 getErrorEy . . . . .	15
3.2.8 getErrorR . . . . .	15
3.2.9 getFormat . . . . .	15
3.2.10 getFullNameAlgorithm . . . . .	16
3.2.11 getFullNameTestFunction . . . . .	16

3.2.12	getLink	16
3.2.13	getLinkAlgorithm	17
3.2.14	getLinkTestFunction	17
3.2.15	getListOfParameterOptions	17
3.2.16	getListOfVectorParameterOptions	18
3.2.17	getMaxCountOfFitness	18
3.2.18	getMeanEx	19
3.2.19	getMeanEy	19
3.2.20	getMeanOfAllEx	19
3.2.21	getMeanOfAllEy	20
3.2.22	getMeanOfAllR	20
3.2.23	getMeanR	21
3.2.24	getNameAlgorithm	21
3.2.25	getNameOption	21
3.2.26	getNameParameter	22
3.2.27	getNameTestFunction	22
3.2.28	getNamesOfParameters	22
3.2.29	getNumberOfExperiments	23
3.2.30	getNumberOfMeasuring	23
3.2.31	getNumberOfParameters	23
3.2.32	getNumberOfRuns	24
3.2.33	getOptionFromListOfParameterOptions	24
3.2.34	getOptionFromListOfParameterOptionsForTable	25
3.2.35	getParameter	25
3.2.36	getSuccessReading	25
3.2.37	getVarianceOfAllEx	26
3.2.38	getVarianceOfAllEy	26
3.2.39	getVarianceOfAllR	27
3.2.40	getVarianceOfEx	27
3.2.41	getVarianceOfEy	27
3.2.42	getVarianceOfR	28
3.2.43	getVersion	28
3.3	Задание данных в класс	28
3.3.1	addListOfParameterOptions	28
3.3.2	addListOfVectorParameterOptions	29
3.3.3	addNameOption	29
3.3.4	addNameParameter	30
3.3.5	setAuthor	30
3.3.6	setCheckAllCombinations	30

3.3.7	setDate . . . . .	31
3.3.8	setDimensionTestFunction . . . . .	31
3.3.9	setEmail . . . . .	31
3.3.10	setErrorEx . . . . .	32
3.3.11	setErrorEy . . . . .	32
3.3.12	setErrorR . . . . .	32
3.3.13	setFormat . . . . .	33
3.3.14	setFullNameAlgorithm . . . . .	33
3.3.15	setFullNameTestFunction . . . . .	34
3.3.16	setLink . . . . .	34
3.3.17	setLinkAlgorithm . . . . .	34
3.3.18	setLinkTestFunction . . . . .	35
3.3.19	setListOfParameterOptions . . . . .	35
3.3.20	setMaxCountOfFitness . . . . .	35
3.3.21	setMeanEx . . . . .	36
3.3.22	setMeanEy . . . . .	36
3.3.23	setMeanOfAllEx . . . . .	37
3.3.24	setMeanOfAllEy . . . . .	37
3.3.25	setMeanOfAllR . . . . .	37
3.3.26	setMeanR . . . . .	38
3.3.27	setNameAlgorithm . . . . .	38
3.3.28	setNameTestFunction . . . . .	38
3.3.29	setNumberOfExperiments . . . . .	39
3.3.30	setNumberOfListOfVectorParameterOptions . . . . .	39
3.3.31	setNumberOfMeasuring . . . . .	40
3.3.32	setNumberOfParameters . . . . .	40
3.3.33	setNumberOfRuns . . . . .	40
3.3.34	setParameter . . . . .	41
3.3.35	setSuccessReading . . . . .	41
3.3.36	setVarianceOfAllEx . . . . .	41
3.3.37	setVarianceOfAllEy . . . . .	42
3.3.38	setVarianceOfAllR . . . . .	42
3.3.39	setVarianceOfEx . . . . .	43
3.3.40	setVarianceOfEy . . . . .	43
3.3.41	setVarianceOfR . . . . .	43
3.3.42	setVersion . . . . .	44
3.4	Операторы . . . . .	44
3.4.1	operator = . . . . .	44
3.5	Служебные функции . . . . .	45

3.5.1	initializationOfVariables . . . . .	45
3.5.2	memoryAllocation . . . . .	45
3.5.3	memoryDeallocation . . . . .	45

# 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\_OnlyDataOfHarrixOptimizationTesting** — Конструктор. Создает пустой экземпляр.
2. **~HarrixClass\_OnlyDataOfHarrixOptimizationTesting** — Деструктор класса.

### Возвращение данных 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. **getDate** — Получение текста переменной XML\_Date - Дата создания документа.
4. **getDimensionTestFunction** — Получение текста переменной XML\_DimensionTestFunction — Размерность тестовой задачи.
5. **getEmail** — Получение текста переменной XML\_Email - Email автора, чтобы можно было с ним связаться.
6. **getErrorEx** — Получение значения ошибки Ex.
7. **getErrorEy** — Получение значения ошибки Ey.
8. **getErrorR** — Получение значения надежности R.
9. **getFormat** — Получение переменной XML\_Format, то есть возвращает название формата документа.
10. **getFullNameAlgorithm** — Получение текста переменной XML\_Full\_Name\_Algorithm — Полное название алгоритма оптимизации.
11. **getFullNameTestFunction** — Получение текста переменной XML\_Full\_Name\_Test\_Function — Полное название тестовой функции.
12. **getLink** — Получение переменной XML\_Link, то есть возвращает ссылку на описание формата файла.
13. **getLinkAlgorithm** — Получение текста переменной XML\_Link\_Algorithm — Ссылка на описание алгоритма оптимизации.
14. **getLinkTestFunction** — Получение текста переменной XML\_Link\_Test\_Function — Ссылка на описание тестовой функции.

15. **getListOfParameterOptions** — Получение списка вектора названий вариантов параметров алгоритма оптимизации.
16. **getListOfVectorParameterOptions** — Получение списка вектора названий вариантов параметров алгоритма оптимизации — это сборник строк из MatrixOfNameParameters, где объединены столбцы. Получение строки параметров эксперимента из списка вектора названий вариантов параметров алгоритма оптимизации — это сборник строк из MatrixOfNameParameters, где объединены столбцы.
17. **getMaxCountOfFitness** — Получение текста переменной Max\_Count\_Of\_Fitness — Максимальное допустимое число вычислений целевой функции для алгоритма.
18. **getMeanEx** — Получение среднего значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).
19. **getMeanEy** — Получение среднего значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).
20. **getMeanOfAllEx** — Получение значения переменной MeanOfAllEx — среднее значение ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще.
21. **getMeanOfAllEy** — Получение значения переменной MeanOfAllEy — среднее значение ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.
22. **getMeanOfAllR** — Получение значения переменной MeanOfAllR — среднее значение надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.
23. **getMeanR** — Получение среднего значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).
24. **getNameAlgorithm** — Получение текста переменной XML\_Name\_Algorithm - Название алгоритма оптимизации.
25. **getNameOption** — Получение имени параметра алгоритма по его номеру.
26. **getNameParameter** — Получение значения параметра настройки какой-то в виде полного наименования.
27. **getNameTestFunction** — Получение текста переменной XML\_Name\_Test\_Function — Название тестовой функции.
28. **getNamesOfParameters** — Получение списка параметров алгоритма (тип селекции, тип скрещивания).
29. **getNumberOfExperiments** — Получение текста переменной XML\_Number\_Of\_Experiments — Количество комбинаций вариантов настроек.

- 30. **getNumberOfMeasuring** — Получение текста переменной XML\_Number\_Of\_Measuring — Размерность тестовой задачи (длина хромосомы решения).
- 31. **getNumberOfParameters** — Получение текста переменной XML\_Number\_Of\_Parameters — Количество проверяемых параметров алгоритма оптимизации.
- 32. **getNumberOfRuns** — Получение текста переменной XML\_Number\_Of\_Runs — Количество прогонов по которому делается усреднение для эксперимента.
- 33. **getOptionFromListOfParameterOptions** — Получение названия вариантов параметров алгоритма оптимизации.
- 34. **getOptionFromListOfParameterOptionsForTable** — Получение названия вариантов параметров алгоритма оптимизации. Но старается где-то сокращать, а где-то добавлять строки.
- 35. **getParameter** — Получение значения параметра настройки какой-то.
- 36. **getSuccessReading** — Получение текста переменной SuccessReading — Успешно ли прошло считывание.
- 37. **getVarianceOfAllEx** — Получение значения переменной VarianceOfAllEx — дисперсия ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще.
- 38. **getVarianceOfAllEy** — Получение значения переменной VarianceOfAllEy — дисперсия ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.
- 39. **getVarianceOfAllR** — Получение значения переменной VarianceOfAllR — дисперсия надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.
- 40. **getVarianceOfEx** — Получение дисперсии значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).
- 41. **getVarianceOfEy** — Получение дисперсии значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).
- 42. **getVarianceOfR** — Получение дисперсии значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).
- 43. **getVersion** — Получение переменной Version, то есть возвращает версию формата документа.

**Задание данных в класс**



1. **addListOfParameterOptions** — Добавление списка вектора названий вариантов параметров алгоритма оптимизации.
2. **addListOfVectorParameterOptions** — Добавление строки параметров эксперимента из списка вектора названий вариантов параметров алгоритма оптимизации — это сборник строк из `MatrixOfNameParameters`, где объединены столбцы.
3. **addNameOption** — Добавление имени параметра алгоритма.
4. **addNameParameter** — Получение значения параметра настройки какой-то в виде полного наименования.
5. **setAuthor** — Задание текста переменной `XML_Author` - Автор документа.
6. **setCheckAllCombinations** — Задание текста переменной `XML_All_Combinations` — Все ли комбинации вариантов настроек просмотрены: 0 или 1.
7. **setDate** — Задание текста переменной `XML_Date` - Дата создания документа.
8. **setDimensionTestFunction** — Задание текста переменной `XML_DimensionTestFunction` — Размерность тестовой задачи.
9. **setEmail** — Задание текста переменной `XML_Email` - Email автора, чтобы можно было с ним связаться
10. **setErrorEx** — Задание значения ошибки `Ex`.
11. **setErrorEy** — Задание значения ошибки `Ey`.
12. **setErrorR** — Задание значения надежности `R`.
13. **setFormat** — Задание переменной `XML_Format` — название формата документа.
14. **setFullNameAlgorithm** — Задание текста переменной `XML_Full_Name_Algorithm` — Полное название алгоритма оптимизации.
15. **setFullNameTestFunction** — Задание текста переменной `XML_Full_Name_Test_Function` — Полное название тестовой функции.
16. **setLink** — Задание переменной `XML_Link` — ссылка на описание формата файла.
17. **setLinkAlgorithm** — Задание текста переменной `XML_Link_Algorithm` — Ссылка на описание алгоритма оптимизации.
18. **setLinkTestFunction** — Задание текста переменной `XML_Link_Test_Function` — Ссылка на описание тестовой функции.

19. **setListOfParameterOptions** — Задание списка вектора названий вариантов параметров алгоритма оптимизации.
20. **setMaxCountOfFitness** — Задание текста переменной `Max_Count_Of_Fitness` — Максимальное допустимое число вычислений целевой функции для алгоритма.
21. **setMeanEx** — Задание среднего значения ошибки `Ex` по измерениям для настройки (сколько точек было по столько и усредняем).
22. **setMeanEy** — Задание среднего значения ошибки `Ey` по измерениям для настройки (сколько точек было по столько и усредняем).
23. **setMeanOfAllEx** — Задание значения переменной `MeanOfAllEx` - среднее значение ошибок `Ex` алгоритма оптимизации по измерениям по всем измерениям вообще
24. **setMeanOfAllEy** — Задание значения переменной `MeanOfAllEy` - среднее значение ошибок `Ey` алгоритма оптимизации по измерениям по всем измерениям вообще.
25. **setMeanOfAllR** — Задание значения переменной `MeanOfAllR` — среднее значение надежностей `R` алгоритма оптимизации по измерениям по всем измерениям вообще.
26. **setMeanR** — Задание среднего значения надежности `R` по измерениям для настройки (сколько точек было по столько и усредняем).
27. **setNameAlgorithm** — Получение текста переменной `XML_Name_Algorithm` - Название алгоритма оптимизации.
28. **setNameTestFunction** — Задание текста переменной `XML_Name_Test_Function` — Название тестовой функции.
29. **setNumberOfExperiments** — Задание текста переменной `XML_Number_Of_Experiments` — Количество комбинаций вариантов настроек.
30. **setNumberOfListOfVectorParameterOptions** — Задание значения элемента массива `NumberOfListOfVectorParameterOptions`.
31. **setNumberOfMeasuring** — Задание текста переменной `XML_Number_Of_Measuring` — Размерность тестовой задачи (длина хромосомы решения).
32. **setNumberOfParameters** — Задание текста переменной `XML_Number_Of_Parameters` — Количество проверяемых параметров алгоритма оптимизации.
33. **setNumberOfRuns** — Задание текста переменной `XML_Number_Of_Runs` — Количество прогонов по которому делается усреднение для эксперимента.

- 34. **setParameter** — Задание значения параметра настройки какой-то.
- 35. **setSuccessReading** — Задание текста переменной SuccessReading — Успешно ли прошло считывание.
- 36. **setVarianceOfAllEx** — Задание значения переменной VarianceOfAllEx — дисперсия ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще
- 37. **setVarianceOfAllEy** — Задание значения переменной VarianceOfAllEy — дисперсия ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.
- 38. **setVarianceOfAllR** — Задание значения переменной VarianceOfAllR — дисперсия надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.
- 39. **setVarianceOfEx** — Получение дисперсии значения ошибки Ex по измерениям для настройки (сколько точек было по столько и усредняем).
- 40. **setVarianceOfEy** — Получение дисперсии значения ошибки Ey по измерениям для настройки (сколько точек было по столько и усредняем).
- 41. **setVarianceOfR** — Получение дисперсии значения надежности R по измерениям для настройки (сколько точек было по столько и усредняем).
- 42. **setVersion** — Задание переменной XML\_Version — версия формата документа.

## Операторы

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

## Служебные функции

- 1. **initializationOfVariables** — Обнуление переменных. Внутренняя функция.
- 2. **memoryAllocation** — Выделяет память под необходимые массивы.
- 3. **memoryDeallocation** — Удаляет память из-под массивов. Внутренняя функция.

## 3 Функции

### 3.1 \_Конструкторы и деструкторы

#### 3.1.1 HarrixClass\_OnlyDataOfHarrixOptimizationTesting

Конструктор. Создает пустой экземпляр.

Код 1. Синтаксис

```
HarrixClass_OnlyDataOfHarrixOptimizationTesting();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

#### 3.1.2 ~HarrixClass\_OnlyDataOfHarrixOptimizationTesting

Деструктор класса.

Код 2. Синтаксис

```
~HarrixClass_OnlyDataOfHarrixOptimizationTesting();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.2 Возвращение данных из класса

#### 3.2.1 getAuthor

Получение текста переменной XML\_Author - Автор документа.

Код 3. Синтаксис

```
QString getAuthor();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.2 getCheckAllCombinations

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

Код 4. Синтаксис

```
bool getCheckAllCombinations();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.3 getDate

Получение текста переменной XML\_Date - Дата создания документа.

Код 5. Синтаксис

```
QString getDate();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.4 getDimensionTestFunction

Получение текста переменной XML\_DimensionTestFunction — Размерность тестовой задачи.

#### Код 6. Синтаксис

```
qint64 getDimensionTestFunction();
```

##### **Входные параметры:**

Отсутствуют.

##### **Возвращаемое значение:**

Значение переменной из описания.

### 3.2.5 getEmail

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

#### Код 7. Синтаксис

```
QString getEmail();
```

##### **Входные параметры:**

Отсутствуют.

##### **Возвращаемое значение:**

Значение переменной из описания.

### 3.2.6 getErrorEx

Получение значения ошибки Ex.

#### Код 8. 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.

### 3.2.7 getErrorEy

Получение значения ошибки Ey.

Код 9. 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.

### 3.2.8 getErrorR

Получение значения надежности R.

Код 10. 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.

### 3.2.9 getFormat

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

Код 11. Синтаксис

```
QString getFormat();
```

#### Входные параметры:

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.10 getFullNameAlgorithm

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

Код 12. Синтаксис

```
QString getFullNameAlgorithm();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.11 getFullNameTestFunction

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

Код 13. Синтаксис

```
QString getFullNameTestFunction();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.12 getLink

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

Код 14. Синтаксис

```
QString getLink();
```



**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.13 getLinkAlgorithm

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

Код 15. Синтаксис

```
QString getLinkAlgorithm();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.14 getLinkTestFunction

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

Код 16. Синтаксис

```
QString getLinkTestFunction();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.15 getListOfParameterOptions

Получение списка вектора названий вариантов параметров алгоритма оптимизации.

#### Код 17. Синтаксис

```
QStringList getListOfParameterOptions(int Number_Of_Parameter);
```

##### **Входные параметры:**

Number\_Of\_Parameter — номер параметра.

##### **Возвращаемое значение:**

Список вектора названий вариантов параметров алгоритма оптимизации.

### 3.2.16 getListOfVectorParameterOptions

Получение списка вектора названий вариантов параметров алгоритма оптимизации — это сборник строк из MatrixOfNameParameters, где объединены столбцы. Получение строки параметров эксперимента из списка вектора названий вариантов параметров алгоритма оптимизации — это сборник строк из MatrixOfNameParameters, где объединены столбцы.

#### Код 18. Синтаксис

```
QStringList getListOfVectorParameterOptions();  
QString getListOfVectorParameterOptions(int Number_Of_Experiment);
```

##### **Входные параметры:**

Number\_Of\_Experiment — номер эксперимента.

##### **Возвращаемое значение:**

Значения строки параметров эксперимента.

### 3.2.17 getMaxCountOfFitness

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

#### Код 19. Синтаксис

```
qint64 getMaxCountOfFitness();
```

##### **Входные параметры:**

Отсутствуют.

##### **Возвращаемое значение:**

Значение переменной из описания.

### 3.2.18 getMeanEx

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

Код 20. Синтаксис

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

#### Входные параметры:

Number\_Of\_Experiment — номер комбинации вариантов настроек.

#### Возвращаемое значение:

Значения среднего значения Ex.

### 3.2.19 getMeanEy

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

Код 21. Синтаксис

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

#### Входные параметры:

Number\_Of\_Experiment — номер комбинации вариантов настроек.

#### Возвращаемое значение:

Значения среднего значения Ey.

### 3.2.20 getMeanOfAllEx

Получение значения переменной MeanOfAllEx — среднее значение ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще

Код 22. Синтаксис

```
double getMeanOfAllEx();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Среднее значение ошибок  $E_x$  алгоритма оптимизации по измерениям по всем измерениям вообще

**3.2.21 getMeanOfAllEy**

Получение значения переменной `MeanOfAllEy` — среднее значение ошибок  $E_y$  алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 23. Синтаксис

```
double getMeanOfAllEy();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Среднее значение ошибок  $E_y$  алгоритма оптимизации по измерениям по всем измерениям вообще.

**3.2.22 getMeanOfAllR**

Получение значения переменной `MeanOfAllR` — среднее значение надежностей  $R$  алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 24. Синтаксис

```
double getMeanOfAllR();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Среднее значение надежностей  $R$  алгоритма оптимизации по измерениям по всем измерениям вообще

### 3.2.23 getMeanR

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

Код 25. Синтаксис

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

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения среднего значения R.

### 3.2.24 getNameAlgorithm

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

Код 26. Синтаксис

```
QString getNameAlgorithm();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.25 getNameOption

Получение имени параметра алгоритма по его номеру.

Код 27. Синтаксис

```
QString getNameOption(int Number_Of_Parameter);
```

**Входные параметры:**

Number\_Of\_Parameter — номер параметра.

**Возвращаемое значение:**

Значения параметра в виде наименования.

### 3.2.26 getNameParameter

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

Код 28. 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.

### 3.2.27 getNameTestFunction

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

Код 29. Синтаксис

```
QString getNameTestFunction();
```

#### Входные параметры:

Отсутствуют.

#### Возвращаемое значение:

Значение переменной из описания.

### 3.2.28 getNamesOfParameters

Получение списка параметров алгоритма (тип селекции, тип скрещивания).

Код 30. Синтаксис

```
QStringList getNamesOfParameters();
```

#### Входные параметры:

Отсутствуют.

**Возвращаемое значение:**

Список параметров алгоритма.

**3.2.29 getNumberOfExperiments**

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

Код 31. Синтаксис

```
qint64 getNumberOfExperiments();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

**3.2.30 getNumberOfMeasuring**

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

Код 32. Синтаксис

```
qint64 getNumberOfMeasuring();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

**3.2.31 getNumberOfParameters**

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

Код 33. Синтаксис

```
qint64 getNumberOfParameters();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.32 getNumberOfRuns

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

Код 34. Синтаксис

```
qint64 getNumberOfRuns();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.33 getOptionFromListOfParameterOptions

Получение названия вариантов параметров алгоритма оптимизации.

Код 35. Синтаксис

```
QString getOptionFromListOfParameterOptions(int Number_Of_Parameter, int  
Number_Of_Option);
```

**Входные параметры:**

Number\_Of\_Parameter — номер параметра.

Number\_Of\_Option — номер считываемой опции у параметра алгоритма оптимизации.

**Возвращаемое значение:**

Название вариантов параметров алгоритма оптимизации.



### 3.2.34 getOptionFromListOfParameterOptionsForTable

Получение названия вариантов параметров алгоритма оптимизации. Но старается где-то сокращать, а где-то добавлять строки.

Код 36. Синтаксис

```
QString getOptionFromListOfParameterOptionsForTable(int Number_Of_Parameter, int  
    Number_Of_Option);
```

#### Входные параметры:

Number\_Of\_Parameter — номер параметра.

Number\_Of\_Option — номер считываемой опции у параметра алгоритма оптимизации.

#### Возвращаемое значение:

Название вариантов параметров алгоритма оптимизации.

### 3.2.35 getParameter

Получение значения параметра настройки какой-то.

Код 37. 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).

### 3.2.36 getSuccessReading

Получение текста переменной SuccessReading — Успешно ли прошло считывание.

Код 38. Синтаксис

```
bool getSuccessReading();
```

#### Входные параметры:

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

### 3.2.37 `getVarianceOfAllEx`

Получение значения переменной `VarianceOfAllEx` — дисперсия ошибок `Ex` алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 39. Синтаксис

```
double getVarianceOfAllEx();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Дисперсия ошибок `Ex` алгоритма оптимизации по измерениям по всем измерениям вообще.

### 3.2.38 `getVarianceOfAllEy`

Получение значения переменной `VarianceOfAllEy` — дисперсия ошибок `Ey` алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 40. Синтаксис

```
double getVarianceOfAllEy();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Дисперсия ошибок `Ey` алгоритма оптимизации по измерениям по всем измерениям вообще.

### 3.2.39 `getVarianceOfAllR`

Получение значения переменной `VarianceOfAllR` — дисперсия надежностей `R` алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 41. Синтаксис

```
double getVarianceOfAllR();
```

#### **Входные параметры:**

Отсутствуют.

#### **Возвращаемое значение:**

Дисперсия надежностей `R` алгоритма оптимизации по измерениям по всем измерениям вообще.

### 3.2.40 `getVarianceOfEx`

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

Код 42. Синтаксис

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

#### **Входные параметры:**

`Number_Of_Experiment` — номер комбинации вариантов настроек.

#### **Возвращаемое значение:**

Значения дисперсии значения `Ex`.

### 3.2.41 `getVarianceOfEy`

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

Код 43. Синтаксис

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

#### **Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения дисперсии значения  $E_y$ .

### 3.2.42 getVarianceOfR

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

Код 44. Синтаксис

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

**Входные параметры:**

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Значения дисперсии значения надежности R.

### 3.2.43 getVersion

Получение переменной Version, то есть возвращает версию формата документа.

Код 45. Синтаксис

```
QString getVersion();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Значение переменной из описания.

## 3.3 Задание данных в класс

### 3.3.1 addListOfParameterOptions

Добавление списка вектора названий вариантов параметров алгоритма оптимизации.

Код 46. Синтаксис

```
void addListOfParameterOptions(QString Option, int Number_Of_Parameter);
```

**Входные параметры:**

Option — добавляемое название варианта параметра алгоритма.

Number\_Of\_Parameter — номер параметра.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.2 addListOfVectorParameterOptions

Добавление строки параметров эксперимента из списка вектора названий вариантов параметров алгоритма оптимизации — это сборник строк из `MatrixOfNameParameters`, где объединены столбцы.

Код 47. Синтаксис

```
void addListOfVectorParameterOptions(QString Option);
```

**Входные параметры:**

Option — добавляемая строка.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.3 addNameOption

Добавление имени параметра алгоритма.

Код 48. Синтаксис

```
void addNameOption(QString Option);
```

**Входные параметры:**

Option - имени параметра алгоритма.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.4 addNameParameter

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

Код 49. Синтаксис

```
void addNameParameter(QString Name, int Number_Of_Experiment);
```

#### Входные параметры:

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Parameter — номер параметра.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.5 setAuthor

Задание текста переменной XML\_Author - Автор документа.

Код 50. Синтаксис

```
void setAuthor(QString Author);
```

#### Входные параметры:

Author — значение переменной из описания.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.6 setCheckAllCombinations

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

Код 51. Синтаксис

```
void setCheckAllCombinations(bool AllCombinations);
```

#### Входные параметры:

AllCombinations — Все ли комбинации вариантов настроек просмотрены: 0 или 1.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.7 setDate

Задание текста переменной XML\_Date - Дата создания документа.

Код 52. Синтаксис

```
void setDate(QString Date);
```

#### Входные параметры:

Date — Дата создания документа.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.8 setDimensionTestFunction

Задание текста переменной XML\_DimensionTestFunction — Размерность тестовой задачи.

Код 53. Синтаксис

```
void setDimensionTestFunction(qint64 DimensionTestFunction);
```

#### Входные параметры:

DimensionTestFunction — Размерность тестовой задачи.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.9 setEmail

Задание текста переменной XML\_Email - Email автора, чтобы можно было с ним связаться

Код 54. Синтаксис

```
void setEmail(QString Email);
```

#### Входные параметры:

Email - Email автора, чтобы можно было с ним связаться.

**Возвращаемое значение:**

Отсутствуют.

**3.3.10 setErrorEx**

Задание значения ошибки Ex.

Код 55. Синтаксис

```
void setErrorEx(double ErrorEx, int Number_Of_Experiment, int Number_Of_Measuring);
```

**Входные параметры:**

ErrorEx — задаваемое значение ошибки;

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Measuring — номер измерения варианта настроек.

**Возвращаемое значение:**

Отсутствуют.

**3.3.11 setErrorEy**

Задание значения ошибки Ey.

Код 56. Синтаксис

```
void setErrorEy(double ErrorEy, int Number_Of_Experiment, int Number_Of_Measuring);
```

**Входные параметры:**

ErrorEy — задаваемое значение ошибки;

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Measuring — номер измерения варианта настроек.

**Возвращаемое значение:**

Отсутствуют.

**3.3.12 setErrorR**

Задание значения надежности R.



Код 57. Синтаксис

```
void setErrorR(double ErrorR, int Number_Of_Experiment, int Number_Of_Measuring);
```

**Входные параметры:**

ErrorR — задаваемое значение надежности;

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Measuring — номер измерения варианта настроек.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.13 setFormat

Задание переменной XML\_Format — название формата документа.

Код 58. Синтаксис

```
void setFormat(QString Format);
```

**Входные параметры:**

Format — название формата документа.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.14 setFullNameAlgorithm

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

Код 59. Синтаксис

```
void setFullNameAlgorithm(QString FullNameAlgorithm);
```

**Входные параметры:**

FullNameAlgorithm — Полное название алгоритма оптимизации.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.15 setFullNameTestFunction

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

Код 60. Синтаксис

```
void setFullNameTestFunction(QString FullNameTestFunction);
```

#### Входные параметры:

FullNameTestFunction — Полное название тестовой функции.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.16 setLink

Задание переменной XML\_Link — ссылка на описание формата файла.

Код 61. Синтаксис

```
void setLink(QString Link);
```

#### Входные параметры:

Link — ссылка на описание формата файла.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.17 setLinkAlgorithm

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

Код 62. Синтаксис

```
void setLinkAlgorithm(QString LinkAlgorithm);
```

#### Входные параметры:

LinkAlgorithm — Ссылка на описание алгоритма оптимизации.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.18 setLinkTestFunction

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

Код 63. Синтаксис

```
void setLinkTestFunction(QString LinkTestFunction);
```

#### Входные параметры:

LinkTestFunction — Ссылка на описание тестовой функции.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.19 setListOfParameterOptions

Задание списка вектора названий вариантов параметров алгоритма оптимизации.

Код 64. Синтаксис

```
void setListOfParameterOptions(QStringList List, int Number_Of_Parameter);
```

#### Входные параметры:

List — список названий параметров, которым будем заменять текущий список.

Number\_Of\_Parameter — номер параметра.

#### Возвращаемое значение:

Значения параметра в виде наименования.

### 3.3.20 setMaxCountOfFitness

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

Код 65. Синтаксис

```
void setMaxCountOfFitness(qint64 MaxCountOfFitness);
```

**Входные параметры:**

MaxCountOfFitness — Максимальное допустимое число вычислений целевой функции для алгоритма.

**Возвращаемое значение:**

Отсутствуют.

**3.3.21 setMeanEx**

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

Код 66. Синтаксис

```
void setMeanEx(double MeanEx, int Number_Of_Experiment);
```

**Входные параметры:**

MeanEx — значение ошибки;

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Отсутствуют.

**3.3.22 setMeanEy**

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

Код 67. Синтаксис

```
void setMeanEy(double MeanEy, int Number_Of_Experiment);
```

**Входные параметры:**

MeanEy — значение ошибки;

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.23 setMeanOfAllEx

Задание значения переменной MeanOfAllEx - среднее значение ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще

Код 68. Синтаксис

```
void setMeanOfAllEx(double Mean);
```

#### Входные параметры:

Mean — среднее значение ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.24 setMeanOfAllEy

Задание значения переменной MeanOfAllEy - среднее значение ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 69. Синтаксис

```
void setMeanOfAllEy(double Mean);
```

#### Входные параметры:

Mean — среднее значение ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.25 setMeanOfAllR

Задание значения переменной MeanOfAllR — среднее значение надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 70. Синтаксис

```
void setMeanOfAllR(double Mean);
```

#### Входные параметры:

Mean — среднее значение надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.26 setMeanR

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

Код 71. Синтаксис

```
void setMeanR(double MeanR, int Number_Of_Experiment);
```

**Входные параметры:**

MeanR — значение ошибки;

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.27 setNameAlgorithm

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

Код 72. Синтаксис

```
void setNameAlgorithm(QString NameAlgorithm);
```

**Входные параметры:**

NameAlgorithm — Название алгоритма оптимизации

**Возвращаемое значение:**

Отсутствуют.

### 3.3.28 setNameTestFunction

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

Код 73. Синтаксис

```
void setNameTestFunction(QString NameTestFunction);
```

**Входные параметры:**

NameTestFunction — Название тестовой функции.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.29 setNumberOfExperiments

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

Код 74. Синтаксис

```
void setNumberOfExperiments(qint64 NumberOfExperiments);
```

**Входные параметры:**

NumberOfExperiments — Количество комбинаций вариантов настроек.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.30 setNumberOfListOfVectorParameterOptions

Задание значения элемента массива NumberOfListOfVectorParameterOptions.

Код 75. Синтаксис

```
void setNumberOfListOfVectorParameterOptions(double Num, int Number);
```

**Входные параметры:**

Num — значение элемента;

Number — номер элемента.

**Возвращаемое значение:**

Значения параметра в виде наименования.

### 3.3.31 setNumberOfMeasuring

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

Код 76. Синтаксис

```
void setNumberOfMeasuring(qint64 NumberOfMeasuring);
```

#### Входные параметры:

NumberOfMeasuring — Размерность тестовой задачи (длина хромосомы решения).

#### Возвращаемое значение:

Отсутствуют.

### 3.3.32 setNumberOfParameters

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

Код 77. Синтаксис

```
void setNumberOfParameters(qint64 NumberOfParameters);
```

#### Входные параметры:

NumberOfParameters — Количество проверяемых параметров алгоритма оптимизации.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.33 setNumberOfRuns

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

Код 78. Синтаксис

```
void setNumberOfRuns(qint64 NumberOfRuns);
```

#### Входные параметры:

NumberOfRuns — Количество прогонов по которому делается усреднение для эксперимента.



**Возвращаемое значение:**

Отсутствуют.

**3.3.34 setParameter**

Задание значения параметра настройки какой-то.

Код 79. Синтаксис

```
void setParameter(int Parameter, int Number_Of_Experiment, int Number_Of_Parameter);
```

**Входные параметры:**

Parameter — значение параметра в виде числа;

Number\_Of\_Experiment — номер комбинации вариантов настроек;

Number\_Of\_Parameter — номер параметра.

**Возвращаемое значение:**

Отсутствуют.

**3.3.35 setSuccessReading**

Задание текста переменной SuccessReading — Успешно ли прошло считывание.

Код 80. Синтаксис

```
void setSuccessReading(bool Success);
```

**Входные параметры:**

SuccessReading — Успешно ли прошло считывание.

**Возвращаемое значение:**

Отсутствуют.

**3.3.36 setVarianceOfAllEx**

Задание значения переменной VarianceOfAllEx — дисперсия ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще

Код 81. Синтаксис

```
void setVarianceOfAllEx(double Variance);
```

**Входные параметры:**

Variance — дисперсия ошибок Ex алгоритма оптимизации по измерениям по всем измерениям вообще.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.37 setVarianceOfAllEy

Задание значения переменной VarianceOfAllEy — дисперсия ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 82. Синтаксис

```
void setVarianceOfAllEy(double Variance);
```

**Входные параметры:**

Variance — дисперсия ошибок Ey алгоритма оптимизации по измерениям по всем измерениям вообще.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.38 setVarianceOfAllR

Задание значения переменной VarianceOfAllR — дисперсия надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.

Код 83. Синтаксис

```
void setVarianceOfAllR(double Variance);
```

**Входные параметры:**

Variance — дисперсия надежностей R алгоритма оптимизации по измерениям по всем измерениям вообще.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.39 setVarianceOfEx

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

Код 84. Синтаксис

```
void setVarianceOfEx(double Variance, int Number_Of_Experiment);
```

#### Входные параметры:

Variance — значение заносимой дисперсии;

Number\_Of\_Experiment — номер комбинации вариантов настроек.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.40 setVarianceOfEy

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

Код 85. Синтаксис

```
void setVarianceOfEy(double Variance, int Number_Of_Experiment);
```

#### Входные параметры:

Variance — значение заносимой дисперсии;

Number\_Of\_Experiment — номер комбинации вариантов настроек.

#### Возвращаемое значение:

Отсутствуют.

### 3.3.41 setVarianceOfR

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

Код 86. Синтаксис

```
void setVarianceOfR(double Variance, int Number_Of_Experiment);
```

**Входные параметры:**

Variance — значение заносимой дисперсии;

Number\_Of\_Experiment — номер комбинации вариантов настроек.

**Возвращаемое значение:**

Отсутствуют.

### 3.3.42 setVersion

Задание переменной XML\_Version — версия формата документа.

Код 87. Синтаксис

```
void setVersion(QString Version);
```

**Входные параметры:**

Version — версия формата документа.

**Возвращаемое значение:**

Отсутствуют.

## 3.4 Операторы

### 3.4.1 operator =

Оператор присваивания.

Код 88. Синтаксис

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

**Входные параметры:**

B — Другой экземпляр класса, который и копируем.

**Возвращаемое значение:**

Отсутствует.

## 3.5 Служебные функции

### 3.5.1 initializationOfVariables

Обнуление переменных. Внутренняя функция.

Код 89. Синтаксис

```
void initializationOfVariables();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.5.2 memoryAllocation

Выделяет память под необходимые массивы.

Код 90. Синтаксис

```
void memoryAllocation();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.

### 3.5.3 memoryDeallocation

Удаляет память из-под массивов. Внутренняя функция.

Код 91. Синтаксис

```
void memoryDeallocation();
```

**Входные параметры:**

Отсутствуют.

**Возвращаемое значение:**

Отсутствует.