

# Functions\_For\_HarrixClass\_DataOfHarrixOptimizat- ionTesting - 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 Functions\_For\_HarrixClass\_DataOfHarrixOptimizationTesting.cpp.

## Оглавление

<b>1 Введение</b>	<b>3</b>
<b>2 Список функций</b>	<b>4</b>
<b>3 Функции</b>	<b>6</b>
3.1 Блок функций проверки равенства переменных нескольких исследований	6
3.1.1 HCDOHOT_CompareOfDataByWilcoxonW	6
3.1.2 HCDOHOT_CompareOfDataForAuthor	7
3.1.3 HCDOHOT_CompareOfDataForCheckAllCombinations	7
3.1.4 HCDOHOT_CompareOfDataForDate	8
3.1.5 HCDOHOT_CompareOfDataForDimensionTestFunction	9
3.1.6 HCDOHOT_CompareOfDataForEmail	9
3.1.7 HCDOHOT_CompareOfDataForFormat	10
3.1.8 HCDOHOT_CompareOfDataForFullNameAlgorithm	11
3.1.9 HCDOHOT_CompareOfDataForFullNameTestFunction	11
3.1.10 HCDOHOT_CompareOfDataForLink	12
3.1.11 HCDOHOT_CompareOfDataForMaxCountOfFitness	13
3.1.12 HCDOHOT_CompareOfDataForNameAlgorithm	13

3.1.13	HCDOHOT_CompareOfDataForNameTestFunction . . . . .	14
3.1.14	HCDOHOT_CompareOfDataForNumberOfExperiments . . . . .	15
3.1.15	HCDOHOT_CompareOfDataForNumberOfMeasuring . . . . .	15
3.1.16	HCDOHOT_CompareOfDataForNumberOfParameters . . . . .	16
3.1.17	HCDOHOT_CompareOfDataForNumberOfRuns . . . . .	17
3.1.18	HCDOHOT_CompareOfDataForVersion . . . . .	17
3.2	Генерация отчетов . . . . .	18
3.2.1	HCDOHOT_GeneratedAnalysisReportFromFile . . . . .	18
3.2.2	HCDOHOT_GeneratedReportAboutAlgorithmFromDir . . . . .	19
3.2.3	HCDOHOT_GeneratedSimpleReportFromFile . . . . .	19
3.3	Функции по работе с классом . . . . .	20
3.3.1	HCDOHOT_NumberFilesInDir . . . . .	20
3.3.2	HCDOHOT_ReadFilesInDir . . . . .	20
3.3.3	HCDOHOT_ReadFilesOnlyDataInDir . . . . .	21

# 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

### Блок функций проверки равенства переменных нескольких исследований

1. **HCDONOT\_CompareOfDataByWilcoxonW** — Проверяет по критерию Вилкосона два исследования алгоритмов.
2. **HCDONOT\_CompareOfDataForAuthor** — Проверяет равенство авторов исследований.
3. **HCDONOT\_CompareOfDataForCheckAllCombinations** — Проверяет равенство переменной, которая говорит все ли рассмотрены функции в исследованиях.
4. **HCDONOT\_CompareOfDataForDate** — Проверяет равенство дат исследований.
5. **HCDONOT\_CompareOfDataForDimensionTestFunction** — Проверяет равенство размерностей тестовой задачи (длина хромосомы решения) в исследованиях.
6. **HCDONOT\_CompareOfDataForEmail** — Проверяет равенство email авторов исследований.
7. **HCDONOT\_CompareOfDataForFormat** — Проверяет равенство форматов файлов в исследованиях.
8. **HCDONOT\_CompareOfDataForFullNameAlgorithm** — Проверяет равенство полных названий алгоритмов в исследованиях.
9. **HCDONOT\_CompareOfDataForFullNameTestFunction** — Проверяет равенство полных названий тестовых функций в исследованиях.
10. **HCDONOT\_CompareOfDataForLink** — Проверяет равенство ссылок на описание версий формата файла в исследованиях.
11. **HCDONOT\_CompareOfDataForMaxCountOfFitness** — Проверяет равенство максимальных допустимых чисел вычислений целевой функции для алгоритма в исследованиях.
12. **HCDONOT\_CompareOfDataForNameAlgorithm** — Проверяет равенство идентификаторов алгоритмов оптимизации: в данных содержится один и тот же алгоритм или же нет.
13. **HCDONOT\_CompareOfDataForNameTestFunction** — Проверяет равенство идентификаторов тестовых функций в исследованиях.
14. **HCDONOT\_CompareOfDataForNumberOfExperiments** — Проверяет равенство количества комбинаций вариантов настроек в исследованиях.

15. **HCDONOT\_CompareOfDataForNumberOfMeasuring** — Проверяет равенство количества экспериментов для каждого набора параметров алгоритма в исследованиях.
16. **HCDONOT\_CompareOfDataForNumberOfParameters** — Проверяет равенство количества проверяемых параметров алгоритма оптимизации в исследованиях.
17. **HCDONOT\_CompareOfDataForNumberOfRuns** — Проверяет равенство количества прогонов, по которому делается усреднение для эксперимента в исследованиях.
18. **HCDONOT\_CompareOfDataForVersion** — Проверяет равенство версий формата файла в исследованиях.

### Генерация отчетов

1. **HCDONOT\_GeneratedAnalysisReportFromFile** — Генерирует отчет-анализ Latex по алгоритму по файлу \*.hdata.
2. **HCDONOT\_GeneratedReportAboutAlgorithmFromDir** — Генерирует отчет Latex по алгоритму по файлам \*.hdata алгоритма, просматривая все файлы в папке. То, чтобы в папке были файлы только одного алгоритма, вы берете на себя.
3. **HCDONOT\_GeneratedSimpleReportFromFile** — Генерирует простой отчет Latex по алгоритму по файлу \*.hdata.

### Функции по работе с классом

1. **HCDONOT\_NumberFilesInDir** — Подсчитывает число HarrixClass\_DataOfHarrixOptimizationTesting файлов в папке.
2. **HCDONOT\_ReadFilesInDir** — Заполняет массив SeveralData данными из всех файлов \*.hdata из папки.
3. **HCDONOT\_ReadFilesOnlyDataInDir** — Заполняет массив SeveralData данными (только исследования) из всех файлов \*.hdata из папки.

## 3 Функции

### 3.1 Блок функций проверки равенства переменных нескольких исследований

#### 3.1.1 HCDONOT\_CompareOfDataByWilcoxonW

Проверяет по критерию Вилкосона два исследования алгоритмов.

Код 1. Синтаксис

```
int HCDONOT_CompareOfDataByWilcoxonW (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *Data2, double Q);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование;

Q — уровень значимости. Может принимать значения:

- 0.002;
- 0.01;
- 0.02;
- 0.05;
- 0.1;
- 0.2.

#### Возвращаемое значение:

- 2 — уровень значимости выбран неправильно (не из допустимого множества);
- 1 — объемы выборок не позволяют провести проверку при данном уровне значимости (или они не положительные);
- 0 — выборки не однородны при данном уровне значимости;
- 1 — выборки однородны при данном уровне значимости;

### 3.1.2 HCDONOT\_CompareOfDataForAuthor

Проверяет равенство авторов исследований.

Код 2. Синтаксис

```
bool HCDONOT_CompareOfDataForAuthor (HarrixCClass_DataOfHarrixCOptimizationTesting *
    Data1, HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);
bool HCDONOT_CompareOfDataForAuthor (HarrixCClass_DataOfHarrixCOptimizationTesting *
    SeveralData, int N);
bool HCDONOT_CompareOfDataForAuthor (
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

#### Входные параметры в функциях перегрузках:

SeveralData — массив исследований;

N — количество исследований в массиве.

#### Возвращаемое значение:

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.3 HCDONOT\_CompareOfDataForCheckAllCombinations

Проверяет равенство переменной, которая говорит все ли рассмотрены функции в исследованиях.

Код 3. Синтаксис

```
bool HCDONOT_CompareOfDataForCheckAllCombinations (
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);
bool HCDONOT_CompareOfDataForCheckAllCombinations (
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);
bool HCDONOT_CompareOfDataForCheckAllCombinations (
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.4 HCDONOT\_CompareOfDataForDate

Проверяет равенство дат исследований.

Код 4. Синтаксис

```
bool HCDONOT_CompareOfDataForDate (HarrixClass_DataOfHarrixOptimizationTesting *
    Data1, HarrixClass_DataOfHarrixOptimizationTesting *Data2);
bool HCDONOT_CompareOfDataForDate (HarrixClass_DataOfHarrixOptimizationTesting *
    SeveralData, int N);
bool HCDONOT_CompareOfDataForDate (HarrixClass_OnlyDataOfHarrixOptimizationTesting
    *SeveralData, int N);
```

**Входные параметры:**

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.



### 3.1.5 HCDONOT\_CompareOfDataForDimensionTestFunction

Проверяет равенство размерностей тестовой задачи (длина хромосомы решения) в исследованиях.

#### Код 5. Синтаксис

```
bool HCDONOT_CompareOfDataForDimensionTestFunction (  
    HarrixClass_DataOfHarrixOptimizationTesting *Data1,  
    HarrixClass_DataOfHarrixOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForDimensionTestFunction (  
    HarrixClass_DataOfHarrixOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForDimensionTestFunction (  
    HarrixClass_OnlyDataOfHarrixOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

#### Входные параметры в функциях перегрузках:

SeveralData — массив исследований;

N — количество исследований в массиве.

#### Возвращаемое значение:

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.6 HCDONOT\_CompareOfDataForEmail

Проверяет равенство email авторов исследований.

#### Код 6. Синтаксис

```
bool HCDONOT_CompareOfDataForEmail (HarrixClass_DataOfHarrixOptimizationTesting *  
    Data1, HarrixClass_DataOfHarrixOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForEmail (HarrixClass_DataOfHarrixOptimizationTesting *  
    SeveralData, int N);  
bool HCDONOT_CompareOfDataForEmail (  
    HarrixClass_OnlyDataOfHarrixOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.7 HCDONOT\_CompareOfDataForFormat

Проверяет равенство форматов файлов в исследованиях.

Код 7. Синтаксис

```
bool HCDONOT_CompareOfDataForFormat (HarrixClass_DataOfHarrixOptimizationTesting *
    Data1, HarrixClass_DataOfHarrixOptimizationTesting *Data2);
bool HCDONOT_CompareOfDataForFormat (HarrixClass_DataOfHarrixOptimizationTesting *
    SeveralData, int N);
bool HCDONOT_CompareOfDataForFormat (
    HarrixClass_OnlyDataOfHarrixOptimizationTesting *SeveralData, int N);
```

**Входные параметры:**

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.8 HCDONOT\_CompareOfDataForFullNameAlgorithm

Проверяет равенство полных названий алгоритмов в исследованиях.

Код 8. Синтаксис

```
bool HCDONOT_CompareOfDataForFullNameAlgorithm (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForFullNameAlgorithm (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForFullNameAlgorithm (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

#### Входные параметры в функциях перегрузках:

SeveralData — массив исследований;

N — количество исследований в массиве.

#### Возвращаемое значение:

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.9 HCDONOT\_CompareOfDataForFullNameTestFunction

Проверяет равенство полных названий тестовых функций в исследованиях.

Код 9. Синтаксис

```
bool HCDONOT_CompareOfDataForFullNameTestFunction (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForFullNameTestFunction (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForFullNameTestFunction (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.10 HCDONOT\_CompareOfDataForLink

Проверяет равенство ссылок на описание версий формата файла в исследованиях.

Код 10. Синтаксис

```
bool HCDONOT_CompareOfDataForLink (HarrixClass_DataOfHarrixOptimizationTesting *
    Data1, HarrixClass_DataOfHarrixOptimizationTesting *Data2);
bool HCDONOT_CompareOfDataForLink (HarrixClass_DataOfHarrixOptimizationTesting *
    SeveralData, int N);
bool HCDONOT_CompareOfDataForLink (HarrixClass_OnlyDataOfHarrixOptimizationTesting
    *SeveralData, int N);
```

**Входные параметры:**

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.11 HCDONOT\_CompareOfDataForMaxCountOfFitness

Проверяет равенство максимальных допустимых чисел вычислений целевой функции для алгоритма в исследованиях.

Код 11. Синтаксис

```
bool HCDONOT_CompareOfDataForMaxCountOfFitness (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForMaxCountOfFitness (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForMaxCountOfFitness (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

#### Входные параметры в функциях перегрузках:

SeveralData — массив исследований;

N — количество исследований в массиве.

#### Возвращаемое значение:

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.12 HCDONOT\_CompareOfDataForNameAlgorithm

Проверяет равенство идентификаторов алгоритмов оптимизации: в данных содержится один и тот же алгоритм или же нет.

Код 12. Синтаксис

```
bool HCDONOT_CompareOfDataForNameAlgorithm (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForNameAlgorithm (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForNameAlgorithm (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

**Входные параметры:**

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если имена алгоритмов одинаковы.

false — если разные.

**3.1.13 HCDONOT\_CompareOfDataForNameTestFunction**

Проверяет равенство идентификаторов тестовых функций в исследованиях.

**Код 13. Синтаксис**

```
bool HCDONOT_CompareOfDataForNameTestFunction (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForNameTestFunction (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForNameTestFunction (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

**Входные параметры:**

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.14 HCDONOT\_CompareOfDataForNumberOfExperiments

Проверяет равенство количества комбинаций вариантов настроек в исследованиях.

Код 14. Синтаксис

```
bool HCDONOT_CompareOfDataForNumberOfExperiments (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForNumberOfExperiments (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForNumberOfExperiments (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

#### Входные параметры в функциях перегрузках:

SeveralData — массив исследований;

N — количество исследований в массиве.

#### Возвращаемое значение:

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.15 HCDONOT\_CompareOfDataForNumberOfMeasuring

Проверяет равенство количества экспериментов для каждого набора параметров алгоритма в исследованиях.

Код 15. Синтаксис

```
bool HCDONOT_CompareOfDataForNumberOfMeasuring (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForNumberOfMeasuring (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForNumberOfMeasuring (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.16 HCDONOT\_CompareOfDataForNumberOfParameters

Проверяет равенство количества проверяемых параметров алгоритма оптимизации в исследованиях.

Код 16. Синтаксис

```
bool HCDONOT_CompareOfDataForNumberOfParameters (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForNumberOfParameters (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForNumberOfParameters (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

**Входные параметры:**

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.



### 3.1.17 HCDONOT\_CompareOfDataForNumberOfRuns

Проверяет равенство количества прогонов, по которому делается усреднение для эксперимента в исследованиях.

#### Код 17. Синтаксис

```
bool HCDONOT_CompareOfDataForNumberOfRuns (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data1,  
    HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForNumberOfRuns (  
    HarrixCClass_DataOfHarrixCOptimizationTesting *SeveralData, int N);  
bool HCDONOT_CompareOfDataForNumberOfRuns (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

#### Входные параметры в функциях перегрузках:

SeveralData — массив исследований;

N — количество исследований в массиве.

#### Возвращаемое значение:

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

### 3.1.18 HCDONOT\_CompareOfDataForVersion

Проверяет равенство версий формата файла в исследованиях.

#### Код 18. Синтаксис

```
bool HCDONOT_CompareOfDataForVersion (HarrixCClass_DataOfHarrixCOptimizationTesting  
    *Data1, HarrixCClass_DataOfHarrixCOptimizationTesting *Data2);  
bool HCDONOT_CompareOfDataForVersion (HarrixCClass_DataOfHarrixCOptimizationTesting  
    *SeveralData, int N);  
bool HCDONOT_CompareOfDataForVersion (  
    HarrixCClass_OnlyDataOfHarrixCOptimizationTesting *SeveralData, int N);
```

#### Входные параметры:

Data1 — первое исследование;

Data2 — второе исследование.

**Входные параметры в функциях перегрузках:**

SeveralData — массив исследований;

N — количество исследований в массиве.

**Возвращаемое значение:**

true — если исследуемый параметр алгоритмов одинаков.

false — если разные.

## 3.2 Генерация отчетов

### 3.2.1 HCDONOT\_GeneratedAnalysisReportFromFile

Генерирует отчет-анализ Latex по алгоритму по файлу \*.hdata.

Код 19. Синтаксис

```
void HCDONOT_GeneratedAnalysisReportFromFile(QString filename, QString pathForSave
, QString pathForTempHtml);
void HCDONOT_GeneratedAnalysisReportFromFile(QString filename, QString pathForSave
);
```

**Входные параметры:**

filename — путь к файлу, из которого считываем данные.

pathForSave — путь к папке, куда сохраняем Latex файлы.

pathForTempHtml — путь к папке куда сохраняем во время работы функции отчет в виде temp.html.

**Возвращаемое значение:**

Отсутствует.

В папке сохранения должны быть находиться файлы names.tex, packages.tex, styles.tex из проекта <https://github.com/Harrix/HarrixLaTeXDocumentTemplate>. Для отчета в виде html берется проект: <https://github.com/Harrix/HarrixHtmlForQWebView>.

### 3.2.2 HCDONOT\_GeneratedReportAboutAlgorithmFromDir

Генерирует отчет Latex по алгоритму по файлам \*.hdata алгоритма, просматривая все файлы в папке. То, чтобы в папке были файлы только одного алгоритма, вы берете на себя.

Код 20. Синтаксис

```
void HCDONOT_GeneratedReportAboutAlgorithmFromDir(QString path, QString
    pathForSave, QString pathForTempHtml);
void HCDONOT_GeneratedReportAboutAlgorithmFromDir(QString path, QString
    pathForSave);
```

#### Входные параметры:

path — путь к папке, из которой считаем файлы.

pathForSave — путь к папке, куда сохраняем Latex файлы.

pathForTempHtml — путь к папке куда сохраняем во время работы функции отчет в виде temp.html.

#### Возвращаемое значение:

Отсутствует.

В папке сохранения должны быть находиться файлы names.tex, packages.tex, styles.tex из проекта <https://github.com/Harrix/HarrixLaTeXDocumentTemplate>. Для отчета в виде html берется проект: <https://github.com/Harrix/HarrixHtmlForQWebView>.

### 3.2.3 HCDONOT\_GeneratedSimpleReportFromFile

Генерирует простой отчет Latex по алгоритму по файлу \*.hdata.

Код 21. Синтаксис

```
void HCDONOT_GeneratedSimpleReportFromFile(QString filename, QString pathForSave,
    QString pathForTempHtml);
void HCDONOT_GeneratedSimpleReportFromFile(QString filename, QString pathForSave);
```

#### Входные параметры:

filename — путь к файлу, из которого считываем данные.

pathForSave — путь к папке, куда сохраняем Latex файлы.

pathForTempHtml — путь к папке куда сохраняем во время работы функции отчет в виде temp.html.

**Возвращаемое значение:**

Отсутствует.

В папке сохранения должны быть находиться файлы names.tex, packages.tex, styles.tex из проекта <https://github.com/Harrix/HarrixLaTeXDocumentTemplate>. Для отчета в виде html берется проект: <https://github.com/Harrix/HarrixHtmlForQWebView>.

### 3.3 Функции по работе с классом

#### 3.3.1 HCDONOT\_NumberFilesInDir

Подсчитывает число HarrixClass\_DataOfHarrixOptimizationTesting файлов в папке.

Код 22. Синтаксис

```
int HCDONOT_NumberFilesInDir(QString path);
```

**Входные параметры:**

path — путь к папке, из которой считаем файлы.

**Возвращаемое значение:**

Число файлов HarrixClass\_DataOfHarrixOptimizationTesting файлов в папке.

#### 3.3.2 HCDONOT\_ReadFilesInDir

Заполняет массив SeveralData данными из всех файлов \*.hdata из папки.

Код 23. Синтаксис

```
int HCDONOT_ReadFilesInDir(HarrixClass_DataOfHarrixOptimizationTesting *  
    SeveralData, QString path, QString pathForTempHtml);  
int HCDONOT_ReadFilesInDir(HarrixClass_DataOfHarrixOptimizationTesting *  
    SeveralData, QString path);
```

**Входные параметры:**

SeveralData — массив, в который записываем данные.

path — путь к папке, из которой считаем файлы.

pathForTempHtml — путь к папке куда сохраняем во время работы функции отчет в виде temp.html.

**Возвращаемое значение:**

Число файлов HarrixClass\_DataOfHarrixOptimizationTesting файлов в папке.

**Примечание:**

Подсчитать число файлов в папке до вызова этой функции можно функцией HCDOHOT\_NumberFilesInDir.

### 3.3.3 HCDOHOT\_ReadFilesOnlyDataInDir

Заполняет массив SeveralData данными (только исследования) из всех файлов \*.hdata из папки.

**Код 24. Синтаксис**

```
int HCDOHOT_ReadFilesOnlyDataInDir(HarrixClass_OnlyDataOfHarrixOptimizationTesting
    *SeveralData, QString path, QString pathForTempHtml);
int HCDOHOT_ReadFilesOnlyDataInDir(HarrixClass_OnlyDataOfHarrixOptimizationTesting
    *SeveralData, QString path);
```

**Входные параметры:**

SeveralData — массив, в который записываем данные.

path — путь к папке, из которой считаем файлы.

pathForTempHtml — путь к папке куда сохраняем во время работы функции отчет в виде temp.html.

**Возвращаемое значение:**

Число файлов HarrixClass\_DataOfHarrixOptimizationTesting файлов в папке.

**Примечание:**

Подсчитать число файлов в папке до вызова этой функции можно функцией HCDOHOT\_NumberFilesInDir.