

Лабораторная работа №1
Горелов С. Ю. СКБ222

Создано системой Doxygen 1.14.0

1 Алфавитный указатель классов	1
1.1 Классы	1
2 Список файлов	3
2.1 Файлы	3
3 Классы	5
3.1 Структура Soldier	5
3.1.1 Подробное описание	5
3.1.2 Конструктор(ы)	5
3.1.2.1 Soldier() [1/2]	5
3.1.2.2 Soldier() [2/2]	6
3.1.3 Данные класса	6
3.1.3.1 full_name	6
3.1.3.2 job	6
3.1.3.3 salary	6
3.1.3.4 unit	6
4 Файлы	7
4.1 Файл /home/tide/Desktop/programming_methods/lab-1/experiment.cpp	7
4.1.1 Функции	8
4.1.1.1 insertion_sort()	8
4.1.1.2 main()	8
4.1.1.3 merge()	9
4.1.1.4 merge_sort()	9
4.1.1.5 operator<()	9
4.1.1.6 operator<<()	9
4.1.1.7 operator<=()	10
4.1.1.8 operator>()	10
4.1.1.9 operator>=()	10
4.1.1.10 read_csv()	10
4.1.1.11 shaker_sort()	11
4.1.1.12 split()	11
4.1.1.13 write_csv()	11

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Soldier	Строка из датасета	5
-------------------------	------------------------------	-------------------

Глава 2

Список файлов

2.1 Файлы

Полный список файлов.

`/home/tide/Desktop/programming_methods/git-folders/lab-1/experiment.cpp` 7

Глава 3

Классы

3.1 Структура Soldier

Строка из датасета

Открытые члены

- `Soldier` ()=default
- `Soldier` (std::string f, std::string j, std::string u, int s)

Открытые атрибуты

- std::string `full_name`
ФИО
- std::string `job`
Должность
- std::string `unit`
Подразделение
- int `salary`
Зарплата

3.1.1 Подробное описание

Строка из датасета

Представляет собой структуру, содержащую информацию о военнослужащем (ФИО, должность, подразделение, зарплата)

3.1.2 Конструктор(ы)

3.1.2.1 `Soldier()` [1/2]

`Soldier::Soldier ()` [default]

3.1.2.2 Soldier() [2/2]

```
Soldier::Soldier (  
    std::string f,  
    std::string j,  
    std::string u,  
    int s) [inline]
```

3.1.3 Данные класса

3.1.3.1 full_name

```
std::string Soldier::full_name
```

ФИО

3.1.3.2 job

```
std::string Soldier::job
```

Должность

3.1.3.3 salary

```
int Soldier::salary
```

Зарплата

3.1.3.4 unit

```
std::string Soldier::unit
```

Подразделение

Объявления и описания членов структуры находятся в файле:

- `/home/tide/Desktop/programming_methods/git-folders/lab-1/experiment.cpp`

Глава 4

Файлы

4.1 Файл /home/tide/Desktop/programming_methods/git-folders/lab-1/experiment.cpp

```
#include <algorithm>
#include <chrono>
#include <fstream>
#include <iostream>
#include <iterator>
#include <sstream>
#include <string>
#include <tuple>
#include <vector>
#include <cstdlib>
#include <matplotlib/matplotlib.h>
```

Классы

- struct `Soldier`
Строка из датасета

Функции

- bool `operator<` (const `Soldier` &a, const `Soldier` &b)
Перегрузка оператора "<" для сравнения объектов `Soldier`.
- bool `operator>` (const `Soldier` &a, const `Soldier` &b)
Перегрузка оператора ">" для сравнения объектов
- bool `operator<=` (const `Soldier` &a, const `Soldier` &b)
Перегрузка оператора ">" для сравнения объектов
- bool `operator>=` (const `Soldier` &a, const `Soldier` &b)
Перегрузка оператора ">=" для сравнения объектов
- std::vector< std::string > `split` (const std::string &str, char del=' ')
Разделить строку по разделителю
- std::vector< `Soldier` > `read_csv` (const std::string &filename)
Считать датасет военнослужащих

- void `write_csv` (std::string filename, const std::vector< `Soldier` > &data)
Записать вектор данных в .csv файл
- template<typename T>
std::ostream & `operator<<` (std::ostream &os, const std::vector< T > &vec)
Перегрузка оператора<< для вывода контейнера
- template<std::random_access_iterator RandomAccessIterator, class Compare>
void `insertion_sort` (RandomAccessIterator first, RandomAccessIterator last, Compare comp)
Сортировка вставкой
- template<std::random_access_iterator RandomAccessIterator, class Compare>
void `shaker_sort` (RandomAccessIterator first, RandomAccessIterator last, Compare comp)
Шейкер-сортировка
- template<std::random_access_iterator RandomAccessIterator, class Compare>
void `merge` (RandomAccessIterator l_first, RandomAccessIterator l_last, RandomAccessIterator r_first, RandomAccessIterator r_last, Compare comp)
Функция для слияния двух отсортированных массивов при сортировке слиянием
- template<std::random_access_iterator RandomAccessIterator, class Compare>
void `merge_sort` (RandomAccessIterator first, RandomAccessIterator last, Compare comp)
Сортировка слиянием
- std::pair< std::vector< double >, std::vector< double > > `get_time` (int j, std::string algo)
Функция для замера времени работы сортировок
- int `main` ()
основная функция программы

4.1.1 Функции

4.1.1.1 `get_time()`

```
std::pair< std::vector< double >, std::vector< double > > get_time (
    int j,
    std::string algo)
```

Функция для замера времени работы сортировок

Аргументы

j	число датасетов для сортировки
algo	какой алгоритм использовать

Возвращает

пара векторов значений (x,y), x - размеры датасетов, y - соответствующее время сортировки (в сек.) для выбранного алгоритма

4.1.1.2 `insertion_sort()`

```
template<std::random_access_iterator RandomAccessIterator, class Compare>
void insertion_sort (
    RandomAccessIterator first,
    RandomAccessIterator last,
    Compare comp)
```

Сортировка вставкой

Аргументы

first	итератор на начало контейнера
last	итератор на конец контейнера
comp	функция сравнения

4.1.1.3 main()

```
int main ()
```

основная функция программы

Считывание данных из датасетов, замер времени различных сортировок, запись отсортированных данных, постройка графиков

4.1.1.4 merge()

```
template<std::random_access_iterator RandomAccessIterator, class Compare>
void merge (
    RandomAccessIterator l_first,
    RandomAccessIterator l_last,
    RandomAccessIterator r_first,
    RandomAccessIterator r_last,
    Compare comp)
```

Функция для слияния двух отсортированных массивов при сортировке слиянием

Аргументы

l_first	итератор на начало 1-ого контейнера
l_last	итератор на конец 1-ого контейнера
r_first	итератор на начало 2-ого контейнера
r_last	итератор на конец 2-ого контейнера
comp	функция сравнения

4.1.1.5 merge_sort()

```
template<std::random_access_iterator RandomAccessIterator, class Compare>
void merge_sort (
    RandomAccessIterator first,
    RandomAccessIterator last,
    Compare comp)
```

Сортировка слиянием

Аргументы

first	итератор на начало контейнера
last	итератор на конец контейнера
comp	функция сравнения

4.1.1.6 operator<()

```
bool operator< (  
    const Soldier & a,  
    const Soldier & b)
```

Перегрузка оператора "<" для сравнения объектов [Soldier](#).

Сначала сравниваются подразделения, затем ФИО, затем зарплата.

4.1.1.7 operator<<()

```
template<typename T>  
std::ostream & operator<< (  
    std::ostream & os,  
    const std::vector< T > & vec)
```

Перегрузка оператора<< для вывода контейнера

4.1.1.8 operator<=()

```
bool operator<= (  
    const Soldier & a,  
    const Soldier & b)
```

Перегрузка оператора ">" для сравнения объектов

См. также

[Soldier](#)

Сначала сравниваются подразделения, затем ФИО, затем зарплата.

4.1.1.9 operator>()

```
bool operator> (  
    const Soldier & a,  
    const Soldier & b)
```

Перегрузка оператора "<" для сравнения объектов

См. также

[Soldier](#)

Сначала сравниваются подразделения, затем ФИО, затем зарплата.

4.1.1.10 operator>=()

```
bool operator>= (  
    const Soldier & a,  
    const Soldier & b)
```

Перегрузка оператора ">=" для сравнения объектов

См. также

[Soldier](#)

Сначала сравниваются подразделения, затем ФИО, затем зарплата.

4.1.1.11 read_csv()

```
std::vector< Soldier > read_csv (  
    const std::string & filename)
```

Считать датасет военнослужащих

Аргументы

filename	Имя датасета (например, "dataset_1.csv")
----------	--

Возвращает

Вектор объектов

4.1.1.12 shaker_sort()

```
template<std::random_access_iterator RandomAccessIterator, class Compare>  
void shaker_sort (  
    RandomAccessIterator first,  
    RandomAccessIterator last,  
    Compare comp)
```

Шейкер-сортировка

Аргументы

first	итератор на начало контейнера
last	итератор на конец контейнера
comp	функция сравнения

4.1.1.13 split()

```
std::vector< std::string > split (  
    const std::string & str,  
    char del = ' ')
```

Разделить строку по разделителю

Функция позволяет разделить переданную строку (std::string) по разделителю (по умолчанию - пробел). Используется для считывания датасетов (.csv файл)

Аргументы

str	строка, которую нужно разделить
del	разделитель

Возвращает

Разделенную строку, представленную в виде вектора строк

4.1.1.14 write_csv()

```
void write_csv (
    std::string filename,
    const std::vector< Soldier > & data)
```

Записать вектор данных в .csv файл

Аргументы

filename	имя файла
data	вектор объектов