МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е. О. Шумова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

«Классы, конструкторы, деструкторы»

ПО ДИСЦИПЛИНЕ «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4831 |  |  |  | К.А. Корнющенков |
| к |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

Задание:

Изучить принципы создания классов с конструкторами, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.

В работе необходимо реализовать класс в соответствии с вариантом задания и создать приложение. В классе должен быть предусмотрен конструктор для установки начальных значений полей, конструктор по умолчанию, конструктор копирования, деструктор. Поля класса должны иметь спецификатор доступа private. Поля могут быть объявлены, как указатели. При использовании указателей в конструкторе класса должна быть инициализация полей, через оператор new, в деструкторе освобождение памяти, через оператор delete. Доступ к полям осуществляется через public методы. В функции main() создается не менее 3 объектов класса (с использованием всех конструкторов) и осуществляется вызов методов класса.

Определите класс t\_end\_d, которому при его создании передается текущее системное время и дата в виде параметров конструктора. Этот класс должен включать функцию-член, выводящую время и дату на экран. (Для нахождения и вывода на экран этих данных воспользуйтесь стандартной библиотечной функцией.)

Листинг программы:

Main.cpp

#include <iostream>

#include "t\_end\_d.hpp"

using namespace std;

int main() {

//standart

cout << "standart" << endl;

t\_end\_d time1;

time1.print();

//parametrs

cout << endl << "parametrs" << endl;

tm \*t;

time\_t ltime;

time(&ltime);

t = localtime(&ltime);

t\_end\_d time2(t);

time2.print();

//coppy

cout << endl << "coppy" << endl;

t\_end\_d time3(time2);

time3.print();

}

t\_end\_d.cpp

#include "t\_end\_d.hpp"

#include <iostream>

t\_end\_d::t\_end\_d(){

time\_t ltime;

time(&ltime);

t = localtime(&ltime);

}

t\_end\_d::t\_end\_d(tm\* t){

this->t = t;

}

t\_end\_d::t\_end\_d(const t\_end\_d& t){

this->t = t.t;

}

t\_end\_d::~t\_end\_d(){

}

void t\_end\_d::print(){

std::cout << "time " << t->tm\_sec << ":" << t->tm\_min << ":" << t->tm\_hour << std::endl;

std::cout << "date " << t->tm\_mday << "." << t->tm\_mon << "." << t->tm\_year << std::endl;

}

t\_end\_d.hpp

#ifndef t\_end\_d\_hpp

#define t\_end\_d\_hpp

#include <stdio.h>

#include <ctime>

class t\_end\_d{

private:

tm \*t;

public:

t\_end\_d();

t\_end\_d(tm\*);

t\_end\_d(const t\_end\_d& t);

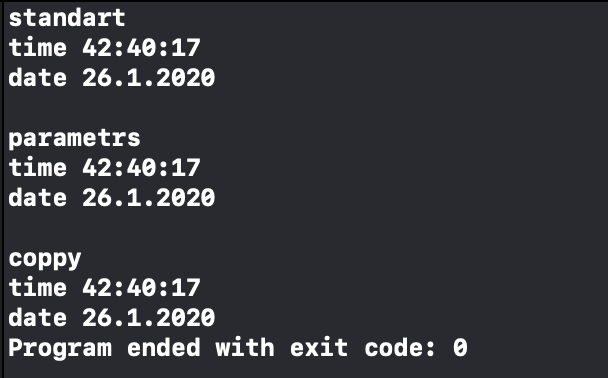
~t\_end\_d();

void print();

};

#endif /\* t\_end\_d\_hpp \*/

Скриншоты с результатами:



Вывод:

В ходе выполнения лабораторной работы был реализован класс t\_end\_d, в котором реализовали 3 вида конструкторов и деструктор.