Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Отчет по лабораторной работе “Классы” №2**

**по дисциплине**

**«Теория алгоритмов и структуры данных»**

Выполнил студент гр. ИВТ-21-1б

Ахунов Руслан Булатович

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

г. Пермь - 2022

**Постановка задачи:**

1.  Определить пользовательский класс.

2.  Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

3.  Определить в классе деструктор.

4.  Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).

5.  Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора-копирования, вызов конструктора с параметрами и конструктора без параметров.

Вариант 2:

Пользовательский класс КНИГА

Название – string

Автор – string

Год издания – int

**Анализ задачи:**

class book

{

private:

    string name;

    string author;

    int year;

public:

    void set(string name, string author, int year)

    {

        this->name = name;

        this->author = author;

        this->year = year;

    }

    void def()

    {

        name = "Дюна";

        author = "Френк Герберт";

        year = 1965;

    }

    void copy(const book& a)

    {

        name = a.name;

        author = a.author;

        year = a.year;

    }

    void print()

    {

        cout << "Название: " << name

            << "\nАвтор: " << author

            << "\nГод издания: " << year << endl;

    }

    ~book()

    {

        cout << "Сработал Деструктор" << endl;

    }

};

Класс book имеет 3 поля в спецификаторе доступа Private: name – Название книги, author – Автор книги, year – Год издания книги.

В Public представлены следующие методы: set – конструктор с параметрами, def – конструктор без параметров, copy – конструктор копирования, print – вывод в консоль данных, ~book – Деструктор.

int main()

{

    setlocale(LC\_ALL, "Russian");

    book a, b, c;

    cout << "Конструктор с параметрами" << endl;

    a.set("Пикник на обочине", "Братья Стругацкие", 1972);

    a.print();

    cout << endl;

    cout << "Конструктор без параметров" << endl;

    b.def();

    b.print();

    cout << endl;

    cout << "Конструктор копирования" << endl;

    c.copy(a);

    c.print();

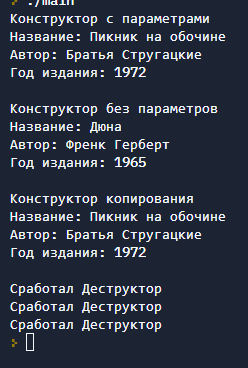
    cout << endl;

    return 0;

}

В функции main мы объявляем 3 переменны типа book – a,b,c. Для переменной a используем конструктор с параметрами, для b – конструктор без параметров, c – конструктор копирования (Копируется переменная a). У всех этих переменных используется метод print.

**Результат работы программы**

****

Вывелось все 3 переменные, в самом конце сработали 3 деструктора, так как завершилась функция main().

**Код программы**

#include <iostream>

#include <string>

using namespace std;

class book

{

private:

    string name;

    string author;

    int year;

public:

    void set(string name, string author, int year)

    {

        this->name = name;

        this->author = author;

        this->year = year;

    }

    void def()

    {

        name = "Дюна";

        author = "Френк Герберт";

        year = 1965;

    }

    void copy(const book& a)

    {

        name = a.name;

        author = a.author;

        year = a.year;

    }

    void print()

    {

        cout << "Название: " << name

            << "\nАвтор: " << author

            << "\nГод издания: " << year << endl;

    }

    ~book()

    {

        cout << "Сработал Деструктор" << endl;

    }

};

int main()

{

    setlocale(LC\_ALL, "Russian");

    book a, b, c;

    cout << "Конструктор с параметрами" << endl;

    a.set("Пикник на обочине", "Братья Стругацкие", 1972);

    a.print();

    cout << endl;

    cout << "Конструктор без параметров" << endl;

    b.def();

    b.print();

    cout << endl;

    cout << "Конструктор копирования" << endl;

    c.copy(a);

    c.print();

    cout << endl;

    return 0;

}

**Вопросы**

1. Для чего нужен конструктор?

Для инициализации объектов класса.

1. Сколько типов конструкторов существует в C++?

3 конструктора: копирования, с параметрами и без параметров.

1. Для чего используется деструктор? В каких случаях деструктор описывается явно?

Применяется для освобождения ресурсов, выделенных конструктором объекту.   
Когда объект содержит указатели на память, выделяемую динамически — иначе при уничтожении объекта память, на которую ссылались его поля-указатели, не будет помечена как свободная.

1. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?

Конструктор без параметров: используется для создания «пустого» объекта;

Конструктор с параметрами: используется для инициализации объекта требуемыми значениями;

Конструктор копирования: для создания копии уже существующего объекта.

1. В каких случаях вызывается конструктор копирования?

Когда в программе необходимо создать копию уже существующего объекта, когда мы передаём объект класса в качестве параметра для функции или когда возвращаем объект класса из функции.

1. Перечислить свойства конструкторов

Конструктор не возвращает значение, даже типа void.

Класс может иметь несколько конструкторов с разными параметрами для разных видов инициализации

Параметры конструктора могут иметь любой тип, кроме этого же класса

Конструкторы не наследуются

Конструкторы нельзя описывать с модификаторами const, virtual и static

Конструкторы глобальных объектов вызываются до вызова функции main

1. Перечислить свойства деструкторов

- не имеет аргументов и возвращаемого значения;

- не наследуется;

- не может быть объявлен как const или static;

- может быть виртуальным;

1. К каким атрибутам имеют доступ методы класса?

Методы класса имеют неограниченный доступ ко всем элементам класса, независимо от спецификаторов доступа и порядка объявления методов в классе.

1. Что представляет собой указатель this?

Указатель на тот объект, для которого он вызван.

1. Какая разница между методами, определенными внутри класса и вне класса?

Определение метода внутри класса ничем не отличается от определения обычной функции. По умолчанию такой метод считается встроенной функцией (inline). Если метод определяется вне функции, то принадлежность метода классу указывается с помощью имени класса: Имя\_класса::Имя\_метода. В классе присутствует только прототип.

1. Какое значение возвращает конструктор?

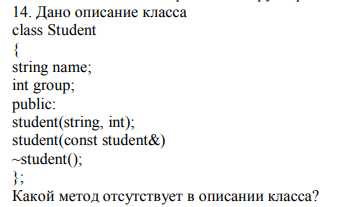
Не возвращает значение, даже типа void

1. Какие методы создаются по умолчанию?

Конструктор без параметров, конструктор копирования и деструктор.

1. Какое значение возвращает деструктор?

Никакое.



Не хватает конструктора без параметров.

https://lh3.googleusercontent.com/HtNjKSY1tmksp_AM2f2sOLHJS8k-6rRTlECUR9Iles8QrlUhnXYwFXKlZOOCuopARl9lnni-SZt1k-4xqRFL9d6BqkOvQru7ZH3obYiFbeA9CP66j5mJcZ2vQlv-gj60b2sJTkZSFC7GELFdn90

Конструктор без параметров.