Фаховий коледж ракетно-космічного машинобудування

Дніпровського національного університету імені Олеся Гончара

ЗВІТ

з лабораторних робіт

з дисципліни «Технології (Програмування)»

Спеціальність 123 Комп’ютерна інженерія

Група КС-21-1

Виконав К.І. Антіпов

Перевірив М.М. Гапоненко

2022-2023

ЗМІСТ

Лабораторна робота № 19

ЛАБОРАТОРНА РОБОТА № 19

Мета: Придбати практичні навички в написанні програми, у якій створюються й руйнуються об'єкти, певного користувачем класу. Виконати дослідження викликів конструкторів і деструкторів.

Хід роботи

1 Постановка задачі.

Загальна постановка завдання:

1.Визначити користувальницький клас відповідно до варіанта завдання (дивися додаток).

2.Визначити в класі наступні конструктори: без параметрів, з параметрами, копіювання.

3.Визначити в класі деструктор.

4.Визначити в класі компонента-функції для перегляду й установки полів даних.

5.Визначити покажчик на компонент-функцію.

6.Визначити покажчик на екземпляр класу.

7.Написати демонстраційну програму, у якій створюються й руйнуються об'єкти користувальницького класу й кожен виклик конструктора й деструктора супроводжується виведенням відповідного повідомлення( який об'єкт який конструктор або деструктор викликав).

8.Показати в програмі використання покажчика на об'єкт і покажчика на компонент-функцію.

3. Описи членів-даних користувальницьких класів

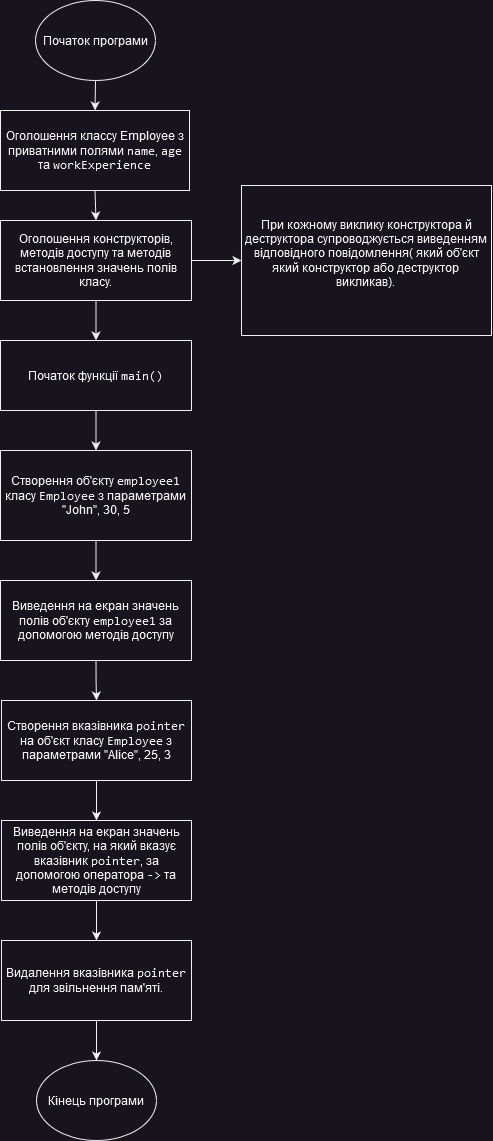
СЛУЖБОВЕЦЬ

ім'я - char\*

вік - іnt

робочий стаж - іnt

2 Блок-схема алгоритму рішення задачі.



3 Текст програми мовою програмування С++.

#include <cstring>

#include <iostream>

#include <windows.h>

using namespace std;

class Employee {

private:

  char \*name; // Поле для збереження імені

  int age;    // Поле для збереження віку

  int workExperience; // Поле для збереження робочого стажу

public:

  // Конструктор за замовчуванням

  Employee() : name(nullptr), age(0), workExperience(0) {

    cout << "Виклик конструктора за замовчуванням" << endl;

  }

  // Параметризований конструктор

  Employee(const char \*name, int age, int workExperience)

      : age(age), workExperience(workExperience) {

    this->name = new char[strlen(name) + 1];

    strcpy(this->name, name);

    cout << "Виклик параметиризованого конструктора за замовчуванням" << endl;

  }

  // Конструктор копіювання

  Employee(const Employee &other)

      : age(other.age), workExperience(other.workExperience) {

    name = new char[strlen(other.name) + 1];

    strcpy(name, other.name);

    cout << "Виклик конструктора за копіюванням" << endl;

  }

  // Деструктор

  ~Employee() {

    delete[] name;

    cout << "Виклик деструктора конструктора" << endl;

  }

  // Методи доступу до полів (геттери)

  const char \*getName() const { return name; }

  int getAge() const { return age; }

  int getWorkExperience() const { return workExperience; }

  // Методи встановлення значень полів (сеттери)

  void setName(const char \*name) {

    delete[] this->name;

    this->name = new char[strlen(name) + 1];

    strcpy(this->name, name);

  }

  void setAge(int age) { this->age = age; }

  void setWorkExperience(int workExperience) {

    this->workExperience = workExperience;

  }

};

int main() {

  SetConsoleCP(65001);       // встановлюємо кодування

  SetConsoleOutputCP(65001); // встановлюємо кодування

  // Приклад використання класу Employee

  Employee employee1("Костя", 17, 5);

  cout << "Ім'я: " << employee1.getName() << ", Вік: " << employee1.getAge()

       << ", Робочий стаж: " << employee1.getWorkExperience() << endl;

  // Використання вказівника на об'єкт

  Employee \*pointer = new Employee("Вася", 25, 3);

  cout << "Ім'я: " << pointer->getName() << ", Вік: " << pointer->getAge()

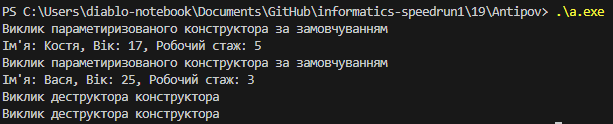
       << ", Робочий стаж: " << pointer->getWorkExperience() << endl;

  delete pointer;

  return 0;

}

4 Копія вікна виконання програми



5 Висновок

В результаті виконання лабораторної роботи опрацьовано теоретичний матеріал за темами:

Класи

Створення показників на методи класу

Створення показників на об’єкти класу

Створення конструкторів та деструкторів класу