Институт информационных технологий

Кафедра: Математическое и программное обеспечение ЭВМ

Дисциплина: Объектно-ориентированное программирование

ЛАБОРАТОРНАЯ РАБОТА № 6

Тема: Наследование

Выполнил:

студент гр. 1ПИб-02-2оп-23

Кринкин Олег Алексеевич

Проверил:

ассистент Матевосян Ремик Артурович

ЗАДАНИЕ

1. Ознакомьтесь с УМП по ООП часть 2 раздел VIII.
2. Создайте базовый класс-интерфейс для АТД, реализованного по заданию ЛР1, состоящий чисто из виртуальных функций, используя одиночное наследование.
3. Разработайте иерархию родственных типов используя простое наследование, количество уровней 4-5 или (5+ классов).
4. В реализации функции вывода (кроме абстрактных классов), необходимо в самом начале обеспечить вывод информации о названии класса (название предметной области), например:

«**Монитор**

Производитель: …

Диагональ: 32 дюйма

….

»

1. Проверьте работоспособность АТД на тестовом наборе данных.

ХОД РАБОТЫ

1. Создана следующая иерархия классов (рис. 1):

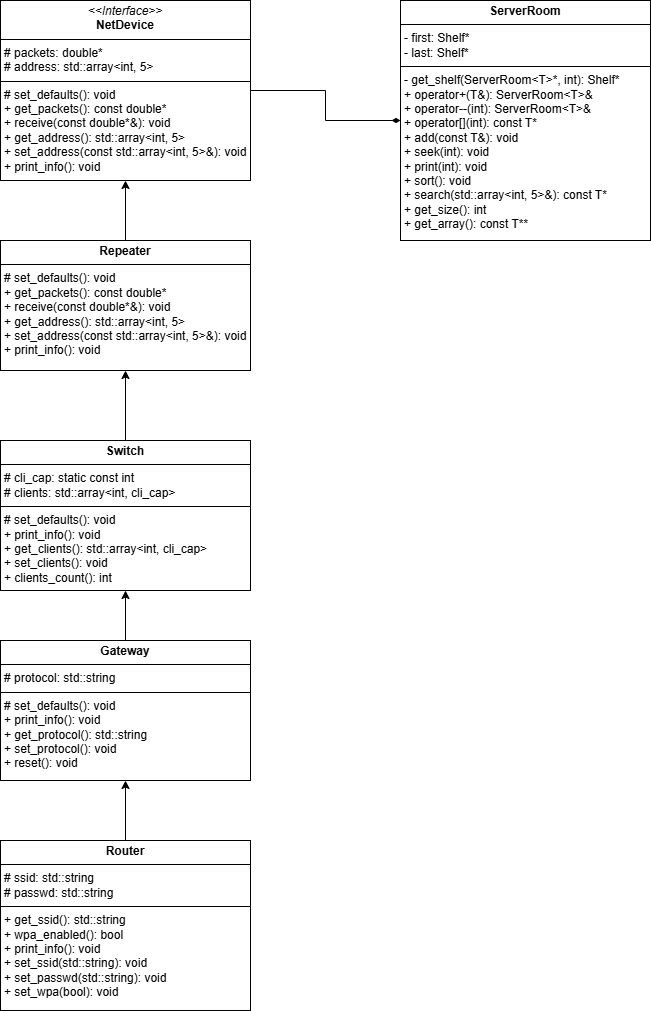


Рис. . Диаграмма классов

1. Был описан интерфейсный класс NetDevice:

|  |
| --- |
| #pragma once  #include <array>  class NetDevice {  protected:  std::array<int, 5> address;  const double\* packets;  virtual void set\_defaults() = 0;  public:  virtual const double\* get\_packets() const = 0;  virtual void receive(const double\*&) = 0;  virtual std::array<int, 5> get\_address() const = 0;  virtual void set\_address(const std::array<int, 5>&) = 0;  virtual void print\_info() const = 0;  }; |

Интерфейсный класс является абстрактным и служит некоторым эталоном для производных классов, описывающий методы и поля, которые обязательно должны быть реализованы во всех производных классах.

1. От него были описаны остальные производные классы. Из уже существующих классов был убран различный повторный код, который стал наследуемым. Так например функция print() в двух разных классах стала выглядеть следующим образом:

|  |
| --- |
| void Router::print\_info() const {  Gateway::print\_info();  std::cout << " - SSID: ";  this->ssid.empty() ? std::cout << this->ssid : std::cout << "None";  std::cout << std::endl;  std::cout << " - WPA: ";  this->wpa ? std::cout << "enabled" : std::cout << "disabled";  std::cout << std::endl;  } |
| void Switch::print\_info() const {  Repeater::print\_info();  std::cout << " - Connected devices (" << this->clients.size() << "): ";  for (int i = 0; i < this->clients.size(); i++)  std::cout << this->clients[i] << " ";  std::cout << std::endl;  } |

Здесь происходит вызов метода из класса-родителя.

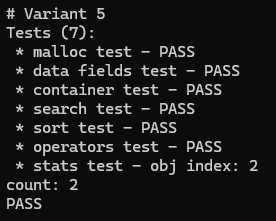
РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ РАБОТЫ

Результат исполнения программы

Используя некоторую систему автоматических «юнит» тестов, которая выполняет несколько тестов:

* выделение и освобождение памяти классами;
* заполнение и хранение данных в полях классов;
* хранение объектов классов в классе-контейнере;
* поиск объекта в контейнере;
* сортировка объектов;
* работа операторов;
* работа статистики.

получен следующий результат:



ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были освоены навыки создания иерархии классов, описания интерфейсного класса. Изучено само понятие интерфейсного класса и получено понимание его назначения.