МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего

образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине

**«Объектно-ориентированное программирование»**

на тему:

**«Наследование в С++»**

*Вариант № 3*

Выполнил:

Студент группы

КТбо2-8

Липов Э. А.

Проверил:

Тарасов С. А.

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Таганрог 2020

# Цель работы

Ознакомление с отношением наследования и иерархией классов.

# Задание, вариант № 3

Создать класс Line (строка). c методами, реализующими операции сложе­ния и вычитания. На его основе реализовать классы Integer и Row. Объект пер­вого рассматривается как обычное целое число, объект второго как строка, для которой сложением является конкатенация, а вычитание двух строк сводится к удалению из первой символов, входящих во вторую строку.

# Ход работы

## Спецификация класса “Line”

Диаграмма класса приведена на Рисунок 1.

Класс “Line” со всеми методами объявлен в заголовочном файле “Line.hpp”. Все функции определены в файле “Line.cpp”.

В классе “Line” определено одно закрытое поле, описывающее данные строки: поле “line\_” (строка).

У Line определен кон­структор “Line(std::string line)”, инициализи­рующий поле, описанные выше.

Для класса определён метод “GetLine()”, дающий воз­можность прочесть закрытое поля класса извне, а также метод “SetLine()”, дающий право записи данных в закрытое поля класса извне.

Text

Description automatically generated with medium confidence

Рисунок — Диаграмма класса “Line”

Также объявлены виртуальные функции для сложения и вычитания (см. Листинг 1.1) экземпляра класса “Line” с другим экземпляром этого класса и со строкой и определён оператор вывода на экран.

* + 1. Листинг 1.1

class Line {

public:

Line() = default;

explicit Line(std::string line);

virtual ~Line() = default;

[[nodiscard]] std::string& GetLine() const;

virtual void Add(const std::string& line) = 0;

virtual void Sub(const std::string& line) = 0;

virtual void Add(const Line& line) = 0;

virtual void Sub(const Line& line) = 0;

virtual void Add(const Line\* line) = 0;

virtual void Sub(const Line\* line) = 0;

friend std::ostream& operator<<(std::ostream& out, const Line& line);

friend std::ostream& operator<<(std::ostream& out, const Line\* line);

protected:

void SetLine(std::string line);

private:

std::string line\_;

};

## Спецификация класса “Integer”

Диаграмма класса приведена на Рисунок 2.

Класс “Integer” является производным класса “Line” и со всеми методами объявлен в заголовочном файле “Integer.hpp”. Все функции определены в файле “Integer.cpp”.

У “Integer” определены конструкторы “Integer()” и “Integer(std::string value)” инициализирующие поле базового класса.

Также в классе были определены виртуальные функции сложения и вычитания из базового класса (см. Листинг 1.2).

Text

Description automatically generated

Рисунок — Диаграмма класса “Integer”

* + 1. Листинг 1.2

class Integer : public Line {

public:

Integer();

explicit Integer(std::string value);

void Add(const std::string& line) override;

void Sub(const std::string& line) override;

void Add(const Line& line) override;

void Sub(const Line& line) override;

void Add(const Line\* line) override;

void Sub(const Line\* line) override;

};

## Спецификация класса “Row”

Диаграмма класса приведена на Рисунок 3.

Text

Description automatically generated

Рисунок — Диаграмма класса “Row”

Класс “Row” является производным класса “Line” и со всеми методами объявлен в заголовочном файле “Row.hpp”. Все функции определены в файле “Row.cpp”.

У “Row” определены конструкторы “Row()” и “Row(std::string value)” инициализирующие поле базового класса.

Также в классе были определены виртуальные функции сложения и вычитания из базового класса (см. Листинг 1.3).

* + 1. Листинг 1.3

class Row : public Line {

public:

Row();

explicit Row(std::string row);

void Add(const std::string& line) override;

void Sub(const std::string& line) override;

void Add(const Line& line) override;

void Sub(const Line& line) override;

void Add(const Line\* line) override;

void Sub(const Line\* line) override;

};

## Спецификация класса “LineFactory”

Диаграмма класса приведена на Рисунок 4.

Graphical user interface, text, application

Description automatically generated

Рисунок — Диаграмма класса “LineFactory”

Класс “LineFactory” со всеми методами объявлен в заголовочном файле “LineFactory.hpp”. Все функции определены в файле “LineFactory.cpp”.

У “LineFactory” определена функция “Create”, возвращающая указатель на созданный объект классов “Integer” или “Row” (см. Листинг 1.4).

* + 1. Листинг 1.4

class LineFactory {

public:

static Line\* Create(const std::string& line, int type = 0);

};

## Спецификация класса “ConsoleActionsFactory”

Диаграмма класса приведена на Рисунок 5.

Text

Description automatically generated

Рисунок — Диаграмма класса “ConsoleActionsFactory”

Класс “ConsoleActionsFactory” со всеми методами объявлен в заголовочном файле “ConsoleActionsFactory.hpp”. Все функции определены в файле “ConsoleActionsFactory.cpp”.

У “ConsoleActionsFactory” определены методы “GetClearConsole()” и “GetWaitForAnyKey()”, возвращающие указатели на функции для очистки консоли и ожидания нажатия клавиши пользователем (см. Листинг 1.5).

* + 1. Листинг 1.5

class ConsoleActionsFactory {

public:

static auto GetClearConsole() -> void (\*)();

static auto GetWaitForAnyKey() -> void (\*)();

};

## Спецификация класса “Console”

Диаграмма класса приведена на Рисунок 6.

Класс “Console” со всеми методами объявлен в заголовочном файле “Console.hpp”. Все функции определены в файле “Console.cpp”.

У “Console” определены метод “Run()” для работы с пользователем, “ShowMenu()” для вывода меню в консоль, “Input(int type, int number)” для ввода из консоли объектов классов “Integer” и “Row”, “LineActions(int action, int type)” для выполнения арифметических действий над созданными объектами.

Text

Description automatically generated

Рисунок — Диаграмма класса “LinkedList”

* + 1. Листинг 1.6

class Console {

public:

void Run();

private:

void ShowMenu();

Line\* Input(int type, int number);

void LineActions(int action, int type = 0);

};

# Диаграмма классов

Диаграмма классов представлена на Рисунок 7.

Diagram

Description automatically generated

Рисунок — Диаграмма классов