МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В.Г. Шухова)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №4

по дисциплине: ООП
тема: «Классы»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
пр. Черников Сергей Викторович

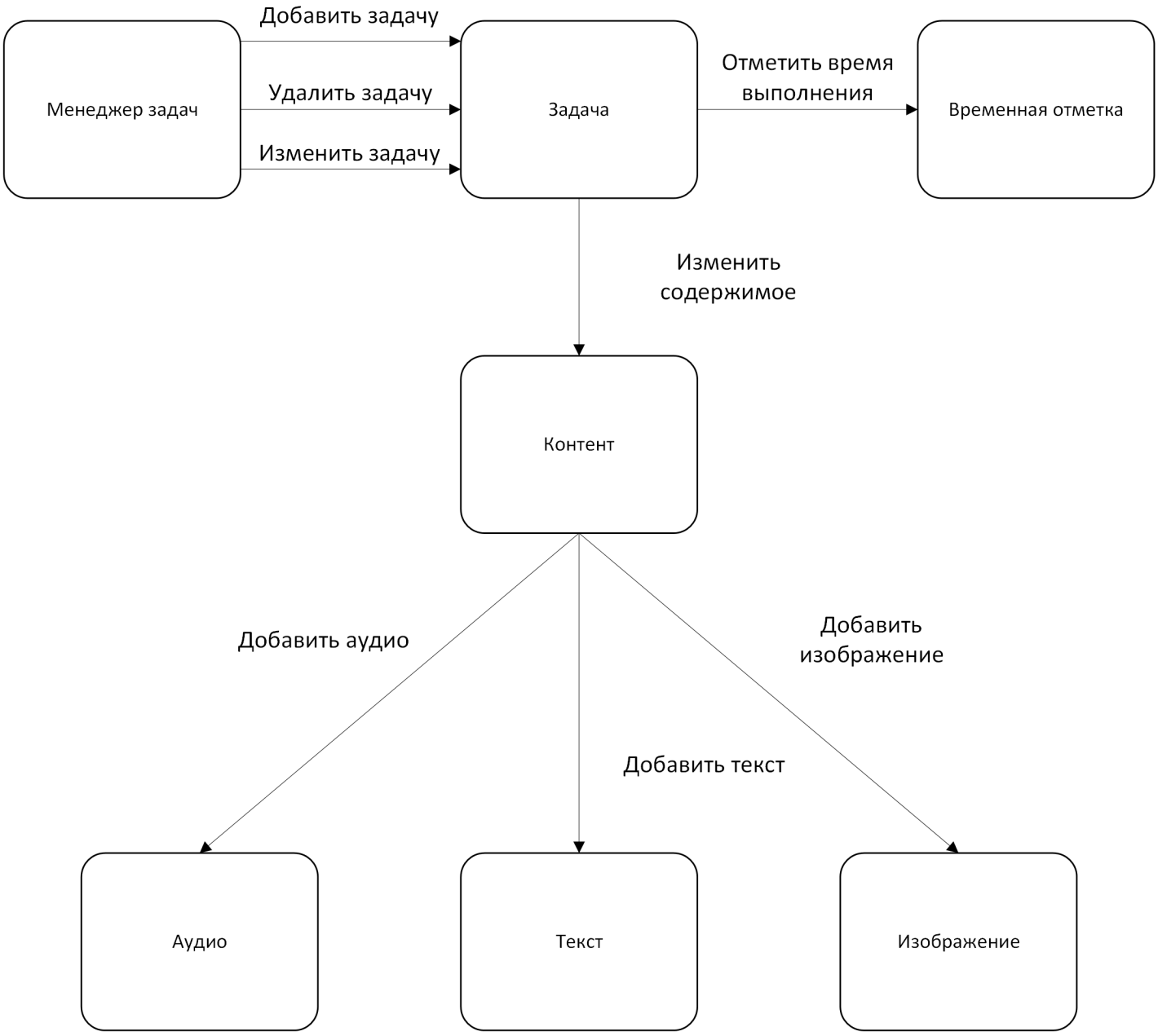
Белгород 2024 г.

Лабораторная работа №4
«Классы»
Вариант 10

Цель работы: приобретение практических навыков создания класса на языке C++.

Задание 1

Выполнить построение диаграммы объектов в соответствии с заданием варианта.
Выполнить построение объектной модели следующей предметной области: “органайзер”.



Задание 2

Создать класс для работы с датой. Разработать следующие элементы класса:

а) Поле DataTime data.

б) Конструкторы, позволяющие установить:
заданную дату;
дату 1.01.2009.

в) Методы, позволяющие: вычислить дату предыдущего дня; вычислить дату следующего дня; определить сколько дней осталось до конца месяца.

г) Перегрузить (переопределить): логический сдвиг влево операция “больше или равно”.

date.h

```
#pragma once

#include <boost/date_time.hpp>

class Date {
    boost::gregorian::date data;
public:
    Date(boost::gregorian::date newDate) : data(newDate) {};
    Date(int d, int m, int y) : data(y, m, d) {};

    Date getDayAfter();
    Date getDayBefore();
    int getDaysBeforeMonthEnd();
    boost::gregorian::date getData();

    Date& operator<<(int amount);
    Date& operator>>(int amount);
    bool operator>=(Date& another) const;
};
```

date.cpp

```
#include "date.h"

Date Date::getDayAfter() {
    return Date(this->data + boost::gregorian::days(1));
}

Date Date::getDayBefore() {
    return Date(this->data - boost::gregorian::days(1));
}

int Date::getDaysBeforeMonthEnd() {
    boost::gregorian::date date_next_month = this->data.end_of_month() + boost::gregorian::days(1);

    return (date_next_month - this->data).days();
}

Date& Date::operator<<(int amount) {
    this->data = this->data - boost::gregorian::days(amount);

    return *this;
}
```

```
Date& Date::operator>>(int amount) {
    this->data = this->data + boost::gregorian::days(amount);

    return *this;
}

bool Date::operator>=(Date& another) const {
    if (this->data.year() < another.data.year()) return false;
    if (this->data.year() > another.data.year()) return true;

    if (this->data.month() < another.data.month()) return false;
    if (this->data.month() > another.data.month()) return true;

    return this->data.day() >= another.data.day();
}

boost::gregorian::date Date::getData() {
    return this->data;
}
```

main.cpp

```
#include <iostream>

#include "../libs/alg/date/date.h"

int main() {
    Date d(23, 11, 2023);

    Date a_lot_of_days_later = d.getDayAfter().getDayAfter().getDayAfter().getDayAfter().getDayAfter().getDayAfter();
    Date a_lot_of_days_before = d.getDayBefore().getDayBefore().getDayBefore().getDayBefore().getDayBefore();

    Date time_travel = d << 1 >> 312 << 23 >> 121 >> 90 << 80;

    std::cout << a_lot_of_days_before.getData() << " " << a_lot_of_days_before.getDaysBeforeMonthEnd() << std::endl;
    std::cout << a_lot_of_days_later.getData() << " " << a_lot_of_days_later.getDaysBeforeMonthEnd() << std::endl;
    std::cout << "Is a_lot_of_days_before >= a_lot_of_days_later? " << (a_lot_of_days_before >= a_lot_of_days_later) <<
    ↪  std::endl;
    std::cout << time_travel.getData() << std::endl;
}
```

Результаты выполнения программы:

```
2023-Nov-18 13
2023-Nov-29 2
Is a_lot_of_days_before >= a_lot_of_days_later? 0
2025-Jan-15
```

Вывод: в ходе лабораторной работы приобрели практические навыки создания класса на языке C++.