МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

Лабораторная работа №1

За 3 семестр

По дисциплине Проектирование программ в интеллектуальных системах

Тема: «Среда программирования С++. Разработка и документирование консольных приложений. Классы»

Выполнил:

Студент 2 курса

Группы ИИ-19

Летченя С. А.

Проверил:

Монтик Н.С

Брест 2021

Цель:

1. Знакомство с интегрированной средой (на примере Microsoft Visual Studio). Изучение ГОСТ 19.701 и правил описания графических схем алгоритмов.

2. Знакомство с разработкой консольных приложений в технологии структурной разработки ПС и перенос задач в среду программирования С++ (например, варианты задач 1 и 2) – тестирование навыков по использованию базовых средств языка С.

3. Изучение индивидуальных заданий по разработке классов, изучение правил описания классов в терминах языка UML, описание классов.

**Ход работы:**

СПИСОК ЗАДАЧ

Получить задания на разработку классов (например, арифметического и не арифметического классов - задачи 4, а, б)

ЗАДАЧА 1 (индивидуальное задание).

Объект автоматизации – предметная область типа “Неарифметический класс”, например, СТУДЕНТ, ГРУППА, ТОВАР, ПРАЙС, КНИГА, РЕЙС, МАРШРУТ и т.п.

Для заданной предметной области следует разработать класс. Он должен в итоговом варианте содержать следующие члены.

1. 2-3 свойства-атрибута (один массивного типа).

2. Три конструктора (по одному каждого типа).

3. Методы, достаточные для установки и получения свойств объекта класса (например, TO\_SET(), TO\_SHOW() и т.п.).

4. Три перегруженных оператора (операторы типа “=”, а также двуместный и одноместный операторы).

ЗАДАЧА 2 (индивидуальное задание).

Объект автоматизации – предметная область типа “Арифметический класс”, например, класс для обработки и хранения трех значений целого типа; класс для обработки и хранения одного значения вещественного и двух значений целого типа; класс для обработки и хранения значения комплексного типа; класс для обработки и хранения значений вектора и т.п.

Для заданной предметной области следует разработать класс аналогично ЗАДАЧЕ 2

#include <iostream>

#include <string>

using namespace std;

class students {

private:

string name\_;

int ball\_rus\_;

int ball\_math\_;

int ball\_phis\_;

public:

students() {

name\_ = "N\A";

ball\_rus\_ = 0;

ball\_math\_ = 0;

ball\_phis\_ = 0;

}

students(string name, int rus, int math, int phis) {

this->name\_ = name;

this->ball\_rus\_ = rus;

this->ball\_math\_ = math;

this->ball\_phis\_ = phis;

}

students(students &anystudents) {

this->name\_ = anystudents.name\_;

this->ball\_rus\_ = anystudents.ball\_rus\_;

this->ball\_math\_ = anystudents.ball\_math\_;

this->ball\_phis\_ = anystudents.ball\_phis\_;

}

students operator++() {

++(this->ball\_rus\_);

++(this->ball\_math\_);

++(this->ball\_phis\_);

return \*this;

}

students operator=(students& anystudents) {

this->name\_ = anystudents.name\_;

this->ball\_rus\_ = anystudents.ball\_rus\_;

this->ball\_math\_ = anystudents.ball\_math\_;

this->ball\_phis\_ = anystudents.ball\_phis\_;

return \*this;

}

students operator-(students& anystudents) {

this->ball\_rus\_ -= anystudents.ball\_rus\_;

this->ball\_math\_ -= anystudents.ball\_math\_;

this->ball\_phis\_ -= anystudents.ball\_phis\_;

return \*this;

}

void show\_score(void) {

cout << "Имя студента: " << this->name\_ << endl;

cout << "Балл по русскому: " << this->ball\_rus\_ << endl;

cout << "Балл по математика: " << this->ball\_math\_ << endl;

cout << "Балл по физика: " << this->ball\_phis\_ << endl;

}

void set\_score(string name, int rus, int math, int phis) {

this->name\_ = name;

this->ball\_rus\_ = rus;

this->ball\_math\_ = math;

this->ball\_phis\_ = phis;

}

void sum\_score(void) {

cout << "Имя студента: " << this->name\_ << endl;

cout << "Общий балл: " << this->ball\_math\_ + this->ball\_phis\_ + this->ball\_rus\_ << endl;

}

};

class business {

private:

string name\_;

int expenses\_;

int profit\_;

public:

business() {

name\_ = "N\A";

expenses\_ = 0;

profit\_ = 0;

}

business(string name, int expenses, int profit) {

this->name\_ = name;

this->expenses\_ = expenses;

this->profit\_ = profit;

}

business(business& anybusiness) {

this->name\_ = anybusiness.name\_;

this->expenses\_ = anybusiness.expenses\_;

this->profit\_ = anybusiness.profit\_;

}

business operator++() {

++(this->expenses\_);

++(this->profit\_);

return \*this;

}

business operator=(business& anybusiness) {

this->name\_ = anybusiness.name\_;

this->expenses\_ = anybusiness.expenses\_;

this->profit\_ = anybusiness.profit\_;

}

business operator-(business& anybusiness) {

this->expenses\_ -= anybusiness.expenses\_;

this->profit\_ -= anybusiness.profit\_;

return \*this;

}

void show\_data(void) {

cout << "Название предприятия: " << this->name\_ << endl;

cout << "Общие затраты: " << this->expenses\_ << endl;

cout << "Общая прибыль: " << this->profit\_ << endl;

}

void set\_data(string name, int expenses, int profit) {

this->name\_ = name;

this->expenses\_ = expenses;

this->profit\_ = profit;

}

};

void main(void) {

setlocale(LC\_ALL, "rus");

system("color 70");

int k;

while (1) {

cout << "Выберите класс(1 либо 2, для выхода 0): " << endl;;

cin >> k;

switch (k) {

case 1: {

business Coke\_Cola;

business Apple("Apple", 6000000000, 3000000);

business Mi(Apple);

Coke\_Cola.show\_data();

Mi.show\_data();

Coke\_Cola.set\_data("Coke-Cola", 321321213, 65465454);

Coke\_Cola.show\_data();

break;

}

case 2: {

students st\_1("Артем", 58, 100, 95);

students st\_2("Иван", 96, 45, 55);

students st\_3("АЛександр", 88, 90, 71);

st\_1.sum\_score();

st\_2.sum\_score();

st\_3.sum\_score();

st\_1 = st\_3;

++st\_1;

st\_1.sum\_score();

break;

}

case 0: {

exit(1);

}

default: {

cout << "Ошибка!" << endl;

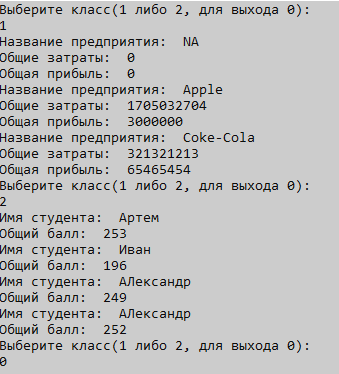
break;

}

}

}

}



Вывод: Изучил работу с классами и успешно применил их на практике.