

Министерство образования Республики Беларусь
Учреждение образования
“Брестский Государственный технический университет”
Кафедра ИИТ

Лабораторная работа №1

По дисциплине “Проектирование программного обеспечения интеллектуальных систем”
Тема: “Среда программирования C++. Разработка и документирование консольных приложений.
Классы”

Выполнил:
Студент 2 курса
Группы ИИ-21
Кирилович А. А.
Проверил:
Монтик Н. С.

Брест 2022

Цель работы:

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 C++ (например, варианты задач 1 и 2) – тестирование навыков по использованию базовых средств языка C.
3. Изучение индивидуальных заданий по разработке классов, изучение правил описания классов в терминах языка UML, описание классов.

Ход работы:

Задание 1

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).

```
#include <iostream>
#include <string>

class Student {
private:
    std::string name;
    std::string surname;
    int dateOfBirthday[3];
public:
    Student() {
        this->name = "NoName";
        this->surname = "NoSurname";

        this->dateOfBirthday[0] = 1;
        this->dateOfBirthday[1] = 1;
        this->dateOfBirthday[2] = 2000;
    }

    Student(std::string name, std::string surname, int date[]) {
        this->name = name;
        this->surname = surname;

        for (int i = 0; i < 3; i++) {
            this->dateOfBirthday[i] = date[i];
        }
    }

    Student(const Student &student) {
        this->name = student.name;
        this->surname = student.surname;

        for (int i = 0; i < 3; i++) {
            this->dateOfBirthday[i] = student.dateOfBirthday[i];
        }
    }

    void setName(std::string name) { this->name = name; }
    void setSurname(std::string surname) { this->surname = surname; }
    void setDateOfBirthday(int date[]) {
        for (int i = 0; i < 3; i++) {
            this->dateOfBirthday[i] = date[i];
        }
    }
}
```

```

    }

    std::string getName() { return this->name; }
    std::string getSurname() { return this->surname; }
    int* getDateOfBirthday() { return this->dateOfBirthday; }
    void operator=(const Student& student) {
        this->name = student.name;
        this->surname = student.surname;

        for (int i = 0; i < 3; i++) {
            this->dateOfBirthday[i] = student.dateOfBirthday[i];
        }
    }
    void operator--(int) {
        this->name = "NoName";
        this->surname = "NoSurname";
        this->dateOfBirthday[0] = 1;
        this->dateOfBirthday[1] = 1;
        this->dateOfBirthday[2] = 2000;
    }

    bool operator==(Student& obj) {
        return this->name == obj.name;
    }
};

int main() {
    Student me;
    std::cout << me.getName() << "\n\n";

    int num[3] = { 9,9,2020 };
    Student T("ars","ki",num);
    std::cout << T.getName() << std::endl;
    T--;
    std::cout << T.getName() << "\n\n";

    std::cout << (me==T);

    return 0;
}

```

Задание 2

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 заданию 1.

```

#include <iostream>
#include <string>

class ComplexNumbers {
private:
    float a;
    float b;
public:
    ComplexNumbers() {
        this->a = 1;
        this->b = 1;
    }

    ComplexNumbers(float a, float b) {
        this->a = a;
        this->b = b;
    }

    ComplexNumbers(const ComplexNumbers& complexNumber) {
        this->a = complexNumber.a;
        this->b = complexNumber.b;
    }
}

```

```

}

void setA(float a) { this->a = a; }
void setB(float b) { this->b = b; }

float getA() { return this->a; }
float getB() { return this->b; }
std::string getComplexNumberAsString() {
    std::string str;
    str = std::to_string(this->a);
    str.append(" + ");
    str.append(std::to_string(this->b));
    str.append("i");

    return str;
}

void operator=(const ComplexNumbers& complexNumber) {
    this->a = complexNumber.a;
    this->b = complexNumber.b;
}

ComplexNumbers operator+(const ComplexNumbers& complexNumber) {
    ComplexNumbers sum;
    sum.a = this->a + complexNumber.a;
    sum.b = this->b + complexNumber.b;
    return sum;
}

ComplexNumbers operator-(const ComplexNumbers& complexNumber) {
    ComplexNumbers difference;
    difference.a = this->a - complexNumber.a;
    difference.b = this->b - complexNumber.b;
    return difference;
}

ComplexNumbers operator*(const ComplexNumbers& complexNumber) {
    ComplexNumbers product;
    product.a = this->a * complexNumber.a - this->b * complexNumber.b;
    product.b = this->a * complexNumber.b + this->b * complexNumber.a;
    return product;
}

ComplexNumbers operator/(const ComplexNumbers& complexNumber) {
    ComplexNumbers quotient;
    quotient.a = (this->a * complexNumber.a + this->b * complexNumber.b) / \
    (complexNumber.a * complexNumber.a + complexNumber.b * complexNumber.b);
    quotient.b = (this->b * complexNumber.a - this->a * complexNumber.b) / \
    (complexNumber.a * complexNumber.a + complexNumber.b * complexNumber.b);
    return quotient;
}

bool operator==(ComplexNumbers& complexNumber) {
    return this->a == complexNumber.a && this->b == complexNumber.b;
}

};

int main() {
    ComplexNumbers a;
    std::cout << a.getA() << " + " << a.getB() << "i" << "\n\n";

    ComplexNumbers b(3,5);
    std::cout << b.getA() << " + " << b.getB() << "i" << "\n\n";

    std::cout << "Sum:" << (a + b).getA() << " + " << (a + b).getB() << "i" << "\n\n";
    std::cout << "Difference:" << (a - b).getA() << " + " << (a - b).getB() << "i" << "\n\n";
    std::cout << "Product:" << (a * b).getA() << " + " << (a * b).getB() << "i" << "\n\n";
    std::cout << "Quotient:" << (a / b).getA() << " + " << (a / b).getB() << "i" << "\n\n";
}

```

Вывод: ознакомился с IDE Microsoft Visual Studio, изучил правила описания классов.