

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А.
БОНЧ-БРУЕВИЧА"

Факультет инфокоммуникационных сетей и систем
Кафедра программной инженерии и вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА №3

«КЛАССЫ»

по дисциплине «ООП»

Вариант №13

Выполнил студент группы
ИКПИ-12
Музычук Артем

1. Постановка задачи

Дополнить систему, состоящую из двух классов COne и CTwo, которые были разработаны в лабораторной работе 2, новым классом CThree. Новый класс должен быть связан public наследованием с классом CTwo. Класс CThree должен иметь одно поле, которое выбирается студентом самостоятельно. Для разрабатываемого класса написать конструкторы умолчания, с параметрами и конструктор копирования, деструктор, методы доступа и метод print(). Написать тестовую программу для проверки работоспособности разработанных классов.

2. Таблицы атрибутов классов

Таблица атрибутов класса COne

N	Назначение	Идентификатор	Секция
1	Число типа long	L (long)	protected
2	Строка типа string	S (string)	protected

Таблица атрибутов класса CTwo

N	Назначение	Идентификатор	Секция
1	Указатель на объект типа COne	P (COne *)	protected
2	Строка типа string	S (string)	protected

Таблица атрибутов класса CThree

N	Назначение	Идентификатор	Секция
1	Указатель на объект типа COne	P (COne *)	protected
2	Строка типа string	S (string)	protected
3	Указатель на массив	a (char *)	protected

3. Описание методов класса Cone

COne() – конструктор по умолчанию

COne(string input , long L) : S(move(input)), L(L) {} – конструктор с аргументами

COne(const COne &arg) – конструктор копирования

COne &operator=(const COne &arg) – оператор копирования

const long getValue() – взятие значения

const string getString() – взятие строки

size_t len() – задание длины

void print() -вывод на экран

~COne() – деструктор

4. Описание методов класса Ctwo

CTwo() – конструктор по умолчанию

CTwo(string s, string ps , int number) : S(move(s)) – конструктор с параметрами

CTwo(const CTwo &arg) – конструктор копирования

CTwo &operator=(const CTwo &arg) - оператор копирования

const COne *getCOne() – обратиться к классу Cone

const string getString() – взятие строки

size_t len() – задание длины

virtual void print() – вывод на экран

~CTwo() – деструктор

5. Описание методов класса Cthree

CThree():CTwo() – конструктор по умолчанию

CThree(string s, string ps, int number) : CTwo(move(s), move(ps), number) – конструктор с параметрами

CThree(const CThree &arg) : CThree(arg.S, arg.P->S, arg.P->L) – конструктор копирования

CThree &operator=(const CThree &arg) – оператор копирования

char &operator[](size_t idx) – оператор копирования

void print() const override – вывод на экран

~CThree() – деструктор

6. Код программы

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

using namespace std;

class COne {
public:
    long L;
    string S;
public:

    COne(){};

    COne(string input = "", long L = 0) : S(move(input)), L(L) {}

    COne(const COne &arg) {
        L = arg.L;
        S = arg.S;
    }

    COne &operator=(const COne &arg) {
        this->L = arg.L;
        this->S = arg.S;
        return *this;
    }

    ~COne() {
        // S.clear();
    };

    long getValue() const {
        return L;
    }

    string getString() const{
        return S;
    }

    size_t len() const {
        return S.size();
    }

    void print() const {
        cout << "\nCOne: " << L << ", " << S ;
    }

    friend class CTwo;
};

class CTwo {
protected:
    string S;
    COne *P;
public:
```

```

CTwo(){}

CTwo(string s = "", string ps = "", int number = 0) : S(move(s)) {
    P = new COne(move(ps), number);
}

CTwo(const CTwo &arg) {
    P = new COne(*arg.P);
    S = arg.S;
}

CTwo &operator=(const CTwo &arg) {
    CTwo temp(arg);
    swap(P, temp.P);
    swap(S, temp.S);
    return *this;
}

~CTwo() {
    delete P;
    S.clear();
}

const COne *getCOne() const {
    return P;
}

const string getString() {
    return S;
}

size_t len() const {
    return S.size();
}

virtual void print() const {
    cout << "CTwo: ";
    if (P) {
        cout << "\"" << S << "\", ";
        P->print();
    } else { cout << "undefined"; }
}
};

class CThree : public CTwo { // ОТНОШЕНИЕ НАСЛЕДОВАНИЯ
public:
    char *a;

public:

CThree(string s, string ps, int number = 0) : CTwo(move(s), move(ps), number) {
    if (number > 0) {
        a = new char[number] {0};
    } else { cout << "undefined"; }
}

CThree(const CThree &arg) : CThree(arg.S, arg.P->S, arg.P->L) {

```

```

    copy(arg.a, arg.a + P->L, a);
}

CThree &operator=(const CThree &arg) {
    CThree temp(arg);
    swap(*this, temp);
    return *this;
}

~CThree() {
    delete[] a;
}

char &operator[](size_t idx) {
    return a[idx];
}

void print() const override {
    cout << "CThree: ";

    if (a) {
        cout << "[";
        for (size_t i = 0; i < 11; i++) {
            cout << a[i];
        }
        cout << "]\n";
    }
    CTwo::print();
}
};

int main() {
    int n, N=11;
    string *s1= new string;
    string *s2= new string;

    cout<<"Введите ваши данные\n ";
    cout<<"Имя: ";
    cin>>*s1;
    cout<<"Фамилия: ";
    cin>>*s2;
    cout<<"Возраст: "; cin>>n;
    CThree third(*s1,*s2,n);
    cout<<"Номер телефона: ";

    for (size_t i = 0; i < N; ++i) cin>>third[i];

    cout<<"\n_____\n"<<endl;
    cout<<"\t[Результат работы программы]\n"<<endl;
    cout << "\nСодержимое объекта класса\n ",third.print(),cout<< endl;
    cout<<"\n_____\n"<<endl;

    cout << "Имя и Фамилия: " << third.getString() << ' ' << third.getCOne()->getString() << endl;
    cout << "Возраст: " << third.getCOne()->getValue() << endl;
    cout << "Телефон: " << third.a << endl;

    // Копирование конструктором

```

```
cout << "\n[Осуществив копирование с помощью конструктора, создадим новый объект класса]\n" << endl;
CThree forth(third);
cout << "third: ", third.print(), cout << endl << endl;
cout << "forth: ", forth.print(), cout << endl;
return 0;
}
```

7. Тесты:

//Введенные значения

```
Введите ваши данные
Имя: артем
Фамилия: музычук
Возраст: 19
Номер телефона: 89621216090
```

//Результат работы программы

[Результат работы программы]

```
Содержимое объекта класса
CThree: [89621216090]
CTwo: "артем",
COne: 19, музычук
```

```
Имя и Фамилия: артем музычук
Возраст: 19
Телефон: 89621216090
```

[Осуществив копирование с помощью конструктора, создадим новый объект класса]

```
third: CThree: [89621216090]
CTwo: "артем",
COne: 19, музычук
```

```
forth: CThree: [89621216090]
CTwo: "артем",
COne: 19, музычук
```

8. Выводы

В ходе лабораторной работы, мы разработали класс CThree, связанный наследованием с классом CTwo. Также написаны тесты, которые показали, что наследование реализовано верно.