

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 №2

«КЛАССЫ»

по дисциплине «ООП»

Вариант №13

Выполнил студент группы
ИКПИ-12
Музычук Артем

1. Постановка задачи

Разработать определения двух класса COne и CTwo, которые связаны отношением включения. Вариант 13- задание 41. Для всех классов требуется написать три вида конструкторов (умолчания, с параметрами и конструктор копирования), деструктор, перегруженный оператор присваивания, методы доступа и метод print(), распечатывающий значения полей объекта. 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

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

void print() – вывод на экран

~CTwo() – деструктор

5. Код программы

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

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

```
    COne(){};
```

```
    COne(string input , long L) : 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();  
    };
```

```
    const long getValue() const {  
        return L;  
    }
```

```
    const string getString() const
```

```

    {
        return S;
    }

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

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

    friend class CTwo;
};

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

    CTwo(string s, string ps, int number) : 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;
    }

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

    const COne *getCOne() {
        return P;
    }

    const string getString() {
        return S;
    }

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

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

```

```

    } else { cout << "undefined"; }
}
};

int main() {
    int n;
    string *s1=new string;
    cout<<"Введите ваш возраст: ";
    cin>>n;
    cout<<endl;
    string *s2= new string;
    cout<<"Введите ваши данные\n ";

    cout<<"Имя: ";
    cin>>*s1;
    cout<<"Фамилия: ";
    cin>>*s2;

    cout<<"\n_____\n"<<endl;
    cout<<"\t[Результат работы программы]\n"<<endl;

    CTwo second(*s1,*s2,n);
    cout << "\nСодержимое объекта класса\n ",second.print(),cout << endl;

    cout<<"\n_____\n"<<endl;

    cout << "Имя и Фамилия: " << second.getString() << ' ' << second.getCOne()->getString() << endl;

    cout << "Возраст: " << second.getCOne()->getValue() << endl;

    // cout<<"\n-В строке объекта CTwo (имя) "<<second.len()/2<<" символов"<<endl;
    // cout<<"-В строке объекта COne (фамилия)"<< second.getCOne()->len()/2<<" символов"<<endl;

    cout<<"\n_____"<<endl;
    cout<<""<<endl;

    // Копирование конструктором
    cout << "[Осуществив копирование с помощью конструктора, создадим новый объект класса]\n" << endl;
    CTwo third(second);
    cout << "second: ", second.print(), cout << endl;
    cout << "third: ", third.print(), cout << endl;

    // Копирование присваиванием
    cout << "\n_____" << endl;
    cout << "[Осуществив копирование с помощью присваивания, создадим новый объект класса]\n" << endl;
    third = second;
    cout << "second: ", second.print(), cout << endl;
    cout << "third: ", third.print(), cout << endl;

    return 0;
}

```

6. Тесты:

//Введенные значения

```
Введите ваш возраст: 19
```

```
Введите ваши данные
```

```
Имя: артем
```

```
Фамилия: музычук
```

//Результат работы программы

```
[Результат работы программы]
```

```
Содержимое объекта класса
```

```
CTwo: "артем",
```

```
COne: 19, музычук
```

```
Имя и Фамилия: артем музычук
```

```
Возраст: 19
```

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

```
second: CTwo: "артем",
```

```
COne: 19, музычук
```

```
third: CTwo: "артем",
```

```
COne: 19, музычук
```

```
[Осуществив копирование с помощью присваивания, создадим новый объект класса]
```

```
second: CTwo: "артем",
```

```
COne: 19, музычук
```

```
third: CTwo: "артем",
```

```
COne: 19, музычук
```

7. Выводы

В ходе лабораторной работы, мы разработали два класса, связанных отношением включения, а также написали тесты для проверки их работоспособности. Программа выполнена верно.