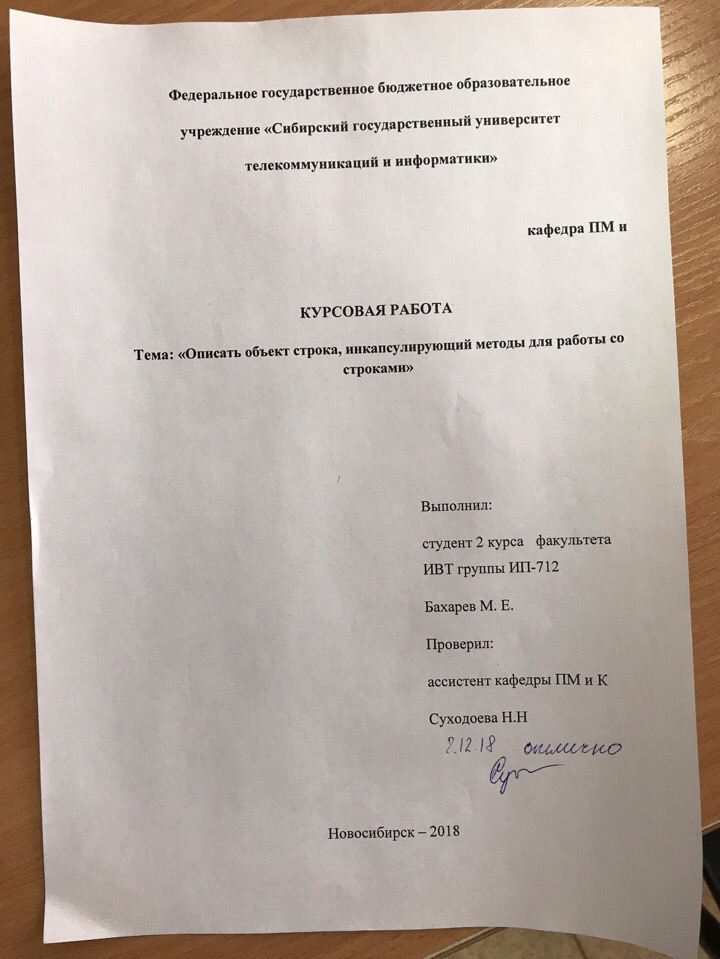
****

**Федеральное государственное бюджетное образовательное**

**учреждение «Сибирский государственный университет**

**телекоммуникаций и информатики»**

**кафедра ПМ и К**

**КУРСОВАЯ РАБОТА**

**Тема: «Описать объект строка, инкапсулирующий методы для работы со строками»**

Выполнил:

студент 2 курса факультета ИВТ группы ИП-712

Бахарев М. Е.

Проверил:

ассистент кафедры ПМ и К

Суходоева Н.Н.

Новосибирск – 2018

Оглавление

[Постановка задачи 3](#_Toc531290180)

[Технологии ООП 4](#_Toc531290181)

[Структура классов 5](#_Toc531290182)

[Алгоритм и результат работы 6](#_Toc531290183)

[Заключение 8](#_Toc531290184)

[Приложение. Листинг 9](#_Toc531290185)

## Постановка задачи

Необходимо реализовать класс String, в котором будут основные методы для работы со строками, такие как очистка, проверка на пустоту, оператор доступа к произвольному символу строки, оператор присваивания. Работа должна быть выполнена с использованием технологий объектно ориентированного программирования (инкапсуляция, наследование, полиморфизм).

# Технологии ООП

В курсовой работе были применены такие принципы объектно-ориентированного программирования, как:

1. Инкапсуляция
2. Полиморфизм
3. Наследование

Из технологий ООП были применены:

1. Дружественные функции
2. Чистые виртуальные функции
3. Абстрактный класс
4. Конструктор, перегрузка конструкторов
5. Конструкторы по умолчанию
6. Списки инициализации
7. Шаблонный класс

Дополнительные применённые технологии:

1. Перегрузка операторов
2. Значения по умолчанию

### Структура классов

В программе реализован абстрактный класс (Container), который содержит в себе два чистых виртуальных метода clear и empty. Предполагается, что в любом контейнере, унаследованном от этого класса эти методы должны определяться.

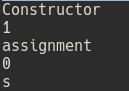
Этот класс является родительским для класса SequenceContainer, который является шаблонным. Этот класс может содержать в себе разные типы данных, а память для них используется последовательно (при помощи массивов). В нём определяются методы clear и empty, а также добавляется оператор доступа к произвольному элементу [ ].

От класса SequenceContainer наследуется класс String, в котором переопределены оператор присваивания и объявлен дружественный оператор вставки, который переопределён за пределами класса.

# Алгоритм и результат работы

В программе имеется три класса: Container, SequenceContainer и String.

В классе Container реализован функционал для классов наследников, которые могут содержать в себе какие-либо элементы стандартных типов (int, char, double и т.д.). В классе SequenceContainer реализован функционал для контейнеров, в которых элементы расположены последовательно в памяти компьютера. В классе String определены методы для работы со строками. В основной части программы происходит демонстрация возможностей класса string: работа конструктора, перегруженный оператор вставки, метод clear, метод empty (для пустой и непустой строки), присваивание, перегруженный оператор доступа к произвольному элементу.

Рис.1

Программа выводит на экран результат своей работы (рис. 1).

# Заключение

При работе использовались такие принципы ООП, как наследование, полиморфизм и инкапсуляция. Наследование – конструирование новых, более сложных производных классов-потомков, из уже имеющихся базовых классов-родителей, с помощью добавления новых полей и/или методов. Полиморфизм - механизм, обеспечивающий возможность определения различных описаний некоторого единого метода (единого по названию) для классов различного уровня иерархии. Инкапсуляция – механизм скрытия данных внутри объекта.

Для создания объектов классов в программе существуют конструкторы, была реализована их перегрузка. Также использовалась дружественная функция, чистые виртуальные методы и абстрактный класс. Дружественные функции – функции, которые не являются компонентом некоторого класса, имеющие доступ ко всем его компонентам. Абстрактный класс – класс, в котором есть хотя бы один чистый (обнуленный) виртуальный метод. Он служит основой для производных классов. Чистые виртуальные функции – функции, которые объявляются в базовом классе как виртуальные и не содержащие описания выполняемых действий.

# Приложение. Листинг

Листинг кода из файла main.cpp:

#include <iostream>  
#include "string.h"  
  
using namespace std;  
  
int main() {  
 String str("Constructor");  
 cout << str << '\n';  
 str.clear();  
 cout << str.empty() << '\n';  
 str = "assignment";  
 cout << str << '\n';  
 cout << str.empty() << '\n';  
 cout << str[2] << endl;  
}

Листинг кода из файла string.h:

#pragma once

#include <ostream>

#include "container.h"

class String : public SequenceContainer<const char> {

public:

String(const char []);

String();

String &operator=(const char arr[]);

friend std::ostream &operator<<(std::ostream &out, const String &str);

};

Листинг кода из файла string.cpp:

#include "string.h"  
  
String::String(const char str[]) : SequenceContainer(str){}  
  
String::String() : SequenceContainer(){}  
  
String &String::operator=(const char \*arr) {  
 this->arr = arr;  
 return \*this;  
}  
  
std::ostream &operator<<(std::ostream &out, const String &str) {  
 if (str.arr) {  
 out << str.arr;  
 }  
 return out;  
}

Листинг кода из файла container.h:

#pragma once

class Container {

public:

virtual void clear() = 0;

virtual bool empty() const = 0;

};

template<typename T>

class SequenceContainer : public Container {

protected:

T \*arr;

public:

SequenceContainer(T \*arr = nullptr) : arr(arr) {};

void clear() override {

arr = nullptr;

}

bool empty() const override {

return !arr;

}

T operator[](int i) {

return arr[i];

};

};