Программирование

Объектно-ориентированное программирование на языке С++

Лабораторная работа №1. Объекты и классы

Теоретический материал.

Перед выполнением работы необходимо проработать раздел 1 пособия [1] ($\S\S$ 1.1–1.3, 1.4.1–1.4.3, 1.5)

Стандартная библиотека С++

Стандартная библиотека является одной из составляющих стандарта C++. Она содержит базовые компоненты для выполнения операций вводавывода, для работы со строками, контейнерами(структурами данных) и алгоритмами.

При использовании модулей и/или библиотек всегда существует вероятность конфликтов имен, которые возникают из-за того, что в модулях и библиотеках один и тот же идентификатор может применяться для разных целей. Для решения этой проблемы в С++ поддерживается концепция пространства имен. Все идентификаторы стандартной библиотеки С++ определяются в пространстве имен с именем **std**. Для использования идентификаторов стандартной библиотеки необходимо записать директиву

using namespace std;

Для стандартной библиотеки C++ используется новая схема именования стандартных заголовочных файлов — стандартные заголовочные файлы не имеют расширения. Команды включения стандартных заголовочных файлов выглядят примерно так:

#include <iostream>

#include <string>

Заголовочные файлы C++ снабжаются префиксом ${\bf c}$ вместо прежнего расширения ${\bf h}$:

#include <cstdlib> // было <stdlib.h #include <cstring> // было <string.h>

Строки в С++

В стандартной библиотеке C++(заголовочный файл **<string>**) определен тип(класс) **string**, который позволяет работать со строками как с обычными типами, не создающими проблем для пользователей. Это означает, что строки можно копировать, присваивать и сравнивать как базовые типы, не беспокоясь о выделении и освобождении памяти. Вы просто используете нужный оператор, например, =(присваивание), ==(проверка на равенство), +(конкатенация).

```
Например.
Определение класса.
class student
string name
                             // имя
int age;
                            // возраст
                            // рейтинг
doudle grade;
public:
student();
                            // конструктор без параметров
student(string,int,double);
                             // конструктор с параметрами
student(const student&); //
                             конструктор копирования
~student();
string getName() const;
int getAge() const;
double getGrade() const;
void setName(string);
void setAge(int);
void setGrade(double);
void set(string,int,double);
void show() const;
};
```

Порядок выполнения работы.

- 1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение).
- 2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
 - 3. Определить в классе деструктор.
 - 4. Определить в классе селекторы и модификаторы
 - 5. Определить указатель на экземпляр класса.
- 6. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).
 - 7. Показать в программе использование указателя на объект.

Методические указания.

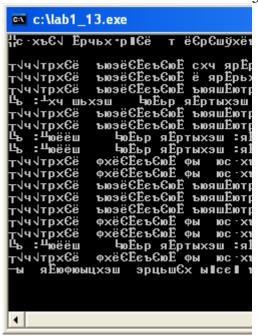
- 1. Следует предусмотреть в программе все возможные способы вызова конструктора копирования. Напоминаем, что конструктор копирования вызывается:
 - а) при использовании объекта для инициализации другого объекта

- б) когда объект передается функции по значению
- в) при построении временного объекта как возвращаемого значения функции
- 2. В программе необходимо предусмотреть размещение объектов, как в статической, так и в динамической памяти, а также создание массивов объектов как в статической, так и в динамической памяти.
 - 3. Программа состоит из трех файлов:
 - * заголовочный h-файл с определением класса;
 - * срр-файл с реализацией класса;
 - * срр-файл демонстрационной программы (функция main()).
- 4. Пример реализации конструктора с выдачей сообщения. student::student(string NAME,int AGE,double GRADE) { name=NAME); age=AGE; grade=GRADE; cout<<"\nКонструктор с параметрами вызван для объекта " <<th>continue this extension of the continue that is a standard of the continue that is a standar
- 4. Для предотвращения многократного включения файла-заголовка следует использовать директивы препроцессора

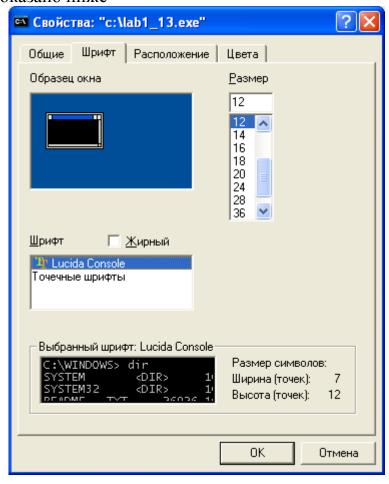
#pragma once

5. Для русификации программы необходимо изменить кодовую страницу. Для этого надо подключить заголовочный файл <windows.h> и в функции main() выполнить SetConsoleOutputCP(1251) и SetConsoleCP(1251).

Если при выполнении программы, Вы получите в окне консоли чтото вроде этого



то следует выбрать в системном меню консольного окна команду "свойства" и установить на вкладке "Шрифт" значение "Lucida Console", как показано ниже



6. Приостановить выполнение программы (пауза) можно функцией system("pause"). Ее прототип – в заголовочном файле <stdlib.h>.

Содержание отчета.

- 1. Титульный лист: название дисциплины; номер и наименование работы; номер варианта; фамилия, имя, отчество студента; дата выполнения.
- 2. Постановка задачи. Следует дать конкретную постановку, т.е. указать, какой класс должен быть реализован, какие должны быть в нем конструкторы, компоненты-функции и т.д.
 - 3. Определение пользовательского класса с комментариями.
 - 4. Реализация конструкторов и деструктора с комментариями.
- 5. Фрагмент программы, показывающий использование указателя на объект с объяснением.
- 6. Листинг основной программы (функция main()), в котором должно быть указано, в каком месте и какой конструктор или деструктор вызываются.

Литература

Ноткин А.М. Технология объектно-ориентированного программирования на языке C++: Электронное учебное пособие — Пермь: Изд-во Перм. гос. техн. ун-та, 2007.

Приложение. Варианты заданий. Описания членов - данных пользовательских классов

1. СТУДЕНТ	2. СЛУЖАЩИЙ	3. КАДРЫ
имя – string	имя – string	имя – string
курс – int	возраст – int	номер цеха – int
пол – bool	должность— string	разряд — int
4. ИЗДЕЛИЕ	5. КНИГА	6. ЭКЗАМЕН
наименование- string	название – string	имя студента – string
шифр – string	автор – string	дата –string
количество – int	стоимость – double	оценка – int
7. АДРЕС	8. TOBAP	9.КВИТАНЦИЯ
имя – string	наименование— string	номер – int
улица — string	количество – int	дата –string
номер дома – int	стоимость – double	сумма – double
10. ЦЕХ	11. ПЕРСОНА	12. АВТОМОБИЛЬ
наименование- string	имя – string	марка – string
начальник – string	возраст – int	мощность – int
количество	пол – bool	стоимость – double
работающих – int		
13. CTPAHA	14. ЖИВОТНОЕ	15. КОРАБЛЬ
имя — string	имя – string	имя – string
форма	класс – string	водоизмещение – int
правления – string	средний вес – int	тип – string
площадь – double		