**Избранные упражнения из книги Стивена Прата «Язык программирования С++»**

Цель проекта: закрепление усвоенного материала из глав посредством практики.

Реализация: проект состоит из главного файла «exercises\_by\_Stephen\_Prata\_C++\_Primer\_Plus.cpp» и 7-ми заголовочных, в которых содержится объявления классов и функций для решения поставленной задачи. Каждый заголовочный файл является отдельным решением для упражнения из главы книги.

Заголовочные файлы хранятся в папке «headers» и содержит следующие файлы:

* «simple\_function\_overloading.h»: упражнение 4 из главы 11 («Работа с классами»), страница 589.

Полное описание из источника:

Перепишите финальный пример класса «Time» так, чтобы все перегруженные операции были реализованы с использованием дружественных функций.

class Time

{

private:

int hours;

int minutes;

public:

Time();

Time(int h, int m = 0);

void AddMin(int m);

void AddHr(int h);

void Reset(int h = 0, int m = 0);

Time operator+(const Time& t) const;

Time operator-(const Time& t) const;

Time operator\* (double n) const;

friend Time operator\*(double m, const Time& t)

{

return t \* m;

} // встроенное определение

friend std::ostream& operator << (std::ostream& os, const Time& t);

};

* «overload\_for\_numbers\_type\_complexxx.h»: упражнение 7 из главы 11 (Работа с классами), страница 589;

Полное описание из источника:

Комплексное число состоит из двух частей - вещественной и мнимой. Один из способов записи такого числа выглядит как (3.0, 4.0). Здесь 3.0 - вещественная часть, а 4. 0 - мнимая. Предположим, что а = (A, Bi) и с = (C, Di). Ниже представлены некоторые операции с комплексными числами:

* + сложение: ;
  + вычитание: ;
  + умножение: ;
  + умножение (х - вещественное число): ;
  + сопряжение: ;

Определите класс complex так, чтобы следующая программа могла использовать его с корректными результатами:

int main()

{

complex a(3.0, 4.0); // инициализация значением (3,4i)

complex c;

cout << "Enter a complex number (q to quit) : \n";

// Ввод комплексного числа (q для завершения)

while (cin >> c)

cout << "c is " << c << '\n'; // значение с

cout << "complex conjugate is " << ~c << '\n';

// значение сопряженного числа

cout << "a is " << a << '\n'; // значение а

cout << "a + c is " << a + c << '\n'; // значение а + с

cout << "a - c is " << a - c << '\n'; // значение а - с

cout << "a \* c is " << a \* c << '\n'; // значение а \* с

cout << "2 .\* c is " << 2 \* c << '\n'; // значение 2 \* с

cout << "Enter a complex number (q to quit) :\n";

cout << "Done!\n";

return 0;

}

Не забывайте, что вы должны перегрузить операции «<<» и «>>». В стандарте С++ уже присутствует поддержка комплексных чисел - и намного более развитая, чем в этом примере - в заголовочном файле complex, поэтому во избежание конфликтов назовите свой файл «complex0.h». Используйте const там, где это оправдано.

* constructor\_overloading.h»: упражнение 1 из главы12 («Классы и динамическое выделение памяти»), страница 657;

Полное описание из источника:

Имеется следующие описание класса:

class Cow

{

private:

char name[20]{};

char\* hobby = new char[10];

double weight = 0.0;

public:

Cow();

Cow(const char\* nm, const char\* ho, double wt);

Cow(const Cow& c);

Cow& operator=(const Cow& ctr1);

void const ShowCow();

~Cow();

};

Напишите реализацию для этого класса и короткую программу, использующую все функции-члены.

* «class\_inheritance.h»: упражнение 1 из главы 13 («Наследование классов»), страница 724;

Полное описание из источника:

Начните со следующего объявления класса:

class Base\_Cd

{

protected:

std::string performer = "noPerformers";

std::string name\_album = "noAlbum";

int num\_of\_songs = 0;

double playtime = 0.0;

public:

Base\_Cd(const std::string&, const std::string&, int, double);

Base\_Cd();

~Base\_Cd();

virtual const void Report();

Base\_Cd& operator=(const Base\_Cd&);

};

Породите класс «Classic», добавив массив членов char, которые будут хранить строку с названием основного произведения на компакт-диске. Если необходимо, чтобы какие-то функции в базовом классе были виртуальными, измените объявление базового класса. Если объявленный метод не нужен, удалите его из определения. Протестируйте результат с помощью следующей программы:

void Bravo(const Cd& disk);

int main()

{

Cd c1("Beatles", "Capitol", 14, 35.5);

Classic c2 = Classic("Piano Sonata in B flat, Fantasia in C",

"Alfred Brendel", "Philips", 2, 57.17);

Cd\* pcd = &c1;

// Непосредственное использование объектов

cout << "Using object directly: \n";

c1.Report(); // использование метода Cd

c2.Report(); // использование метода Classic

// Использование указателя на объекты типа cd \*

cout << "Using type cd \* pointer to objects: \n";

pcd->Report(); // использование метода Cd для объекта cd

pcd = &c2;

pcd->Report(); // использование метода Classic для объекта classic

// Вызов функции с аргументом-ссылкой на Cd

cout << "Calling a function with a Cd reference argument:\n";

Bravo(c1);

Bravo(c2);

// Тестирование присваивания

cout << "Testing assignment: ";

Classic copy;

copy = c2;

copy.Report();

return 0;

}

void Bravo(const Cd& disk)

{

disk.Report();

}

* «multiple\_and\_virtual\_inheritance.h»: упражнение 4 из главы 14 («Повторное использование кода в С++»), страница 801;

Полное описание из источника:

Класс «Person» (Человек) предназначен для хранения имени и фамилии человека. Кроме конструкторов он содержит метод «Show ()» для вывода этих данных.

Класс «Gunslinger» (Снайпер) виртуально порожден от класса «Person». Он содержит член «Draw ()», который возвращает значение типа «double» - время, необходимое снайперу для перехода в боевую готовность. Класс также имеет член типа «int», содержащий количество насечек на винтовке. И, наконец, класс содержит функцию «Show ()», которая выводит всю эту информацию.

Класс «PokerPlayer» (Игрок в покер) виртуально порожден от класса «Person». Он имеет метод «Draw ()», который возвращает случайное число в диапазоне от 1 до 52 - значение карты. (Можно создать класс «Card» с членами, определяющими масть и рубашку карты, чтобы метод «Draw ()» возвращал значение типа «Card».) Класс «PokerPlayer» использует функцию «Show ()» класса «Person».

Класс «BadDude» (Хулиган) открыто порожден от классов «Gunslinger» и «PokerPlayer». Он содержит член «Gdraw ()», возвращающий время вынимания оружия, и член «Cdraw ()», возвращающий следующую вытянутую карту. У него есть соответствующая функция «Show ()».

Определите все упомянутые классы и методы вместе с другими необходимыми методами (такими как методы для задания значений объекта) и протестируйте их с помощью простой программы, подобной представленной в листинге 14.12.

* «palindrome\_test.h»: упражнение 1 из главы 16 («Класс string и стандартная библиотека шаблонов»), страница 964;

Полное описание из источника:

Палиндром — это строка, которая читается одинаково в обоих направлениях. Например, «tot» и «otto» - короткие палиндромы. Напишите программу, которая запрашивает у пользователя строку и передает ссылку на нее функции «boo1». Функция должна возвращать «true», если строка является палиндромом, и «false» - в противном случае. Пока не беспокойтесь о таких нюансах, как прописные и строчные буквы, пробелы и знаки препинания. Другими словами, эта простая версия должна отклонять строки типа «Otto» и «Madam, I'm Adam». Для упрощения решения задачи можете обратиться к списку строковых методов, приведенному в приложении Е.

* «party\_invitation.h»: упражнение 8 из главы 16 («Класс string и стандартная библиотека шаблонов»), страница 965.

Полное описание из источника:

Мэт и Пэт хотят пригласить своих друзей на вечеринку. Они просят вас написать программу, которая делает следующее.

* + Позволяет Мэту ввести список имен его друзей. Имена сохраняются в контейнере и затем отображаются в отсортированном порядке.
  + Позволяет Пэт ввести список ее друзей. Имена сохраняются во втором контейнере и затем отображаются в отсортированном порядке.
  + Создает третий контейнер, который объединяет эти два списка, исключает дубликаты и отображает содержимое этого контейнера.

В папке «files\_for\_display\_on\_screen» содержаться файлы для вывода информации на экран:

* «expected\_result.txt» - храниться построчно примерный итоговый вывод для каждого задания;
* «task\_description.txt» - храниться построчно описание задания для упражнения. При этом для сокращения большого вывода на экран задания немного скорректированы и сокращены с сохранением основной сути.

Ввиду того что книга Стивена Прата «Язык программирования С++» вышла давно (в 2012) было принято решения при программировании использовать стандарт языка «С++14».

Функция «main()» в главном исполняемом файле используется для создания экземпляра класса, в котором объявлен метод для вывода результата работы всех функций-решений к упражнениям. В данной функции последовательно вызывается каждая функция, которая содержит конкретное использование методов для решения каждого упражнения. Каждая такая функция соответствует определенному заголовочному файлу, в котором объявлены методы и классы для решения упражнения.