

Last updated: 2019-04-07

vadimov@i.ua

Практическое занятие 9 (семестр2).

Задача 9.1. Создайте иерархию классов для хранения информации о летательных аппаратах. Начните с общего базового класса **Airship**, предназначенного для хранения количества перевозимых пассажиров и количества перевозимого груза в фунтах или килограммах, или тоннах. Затем создайте два производных от **Airship** класса **Airplane** и **Balloon** (дирижабль). Класс **Airplane** должен хранить тип самолетного двигателя (винтовой или реактивный) и дальность полета в милях или километрах. Класс **Balloon** должен хранить информацию о типе газа, используемого для подъема дирижабля (водород или гелий), и его максимальный потолок (в футах или метрах). Создайте программу для демонстрации работы этой иерархии классов.

Задача 9.2. Напишите программу, которая бы устанавливала флаги для потока **cout** так, чтобы целые, если они положительны, выводились со знаком +. Покажите, что ваш набор флагов формата правилен.

Задача 9.3. Напишите программу, которая бы устанавливала флаги для потока **cout** так, чтобы всегда при выводе дробных значений были показаны десятичные точки. Кроме этого, значения с плавающей точкой должны выводиться в научной нотации с символом "E" в верхнем регистре.

Задача 9.4. Напишите программу, которая сохраняет текущее состояние флагов формата, устанавливает флаги **showbase** и **hex**, выводит на экран значение 100, а затем возвращает флаги в исходное состояние.

Задача 9.5. Разработайте программу для печати таблицы натуральных и десятичных логарифмов чисел от 2 до 100. Формат таблицы следующий: правое выравнивание, ширина поля — 10 символов, точность — 5 десятичных позиций.

Задача 9.6. Создайте функцию `center()` со следующим прототипом:

```
void center(char *s);
```

Эта функция должна устанавливать заданную строку в центр экрана. Для реализации этой задачи воспользуйтесь функцией **width()**. Предполагается, что ширина экрана равна 80 символам. (Для простоты считайте, что длина строки не превышает 80 символов.) Напишите программу, иллюстрирующую работу этой функции.

Совет: Поэкспериментируйте с флагами и функциями формата. После того как вы ближе познакомитесь с системой ввода/вывода C++, вы, хочется верить, никогда не ошибетесь в выборе нужного формата вывода. :)

Задача 9.7. Выполните еще раз Задачу 9.5 и 9.6, только теперь, вместо функций-членов и флагов формата, используйте манипуляторы ввода/вывода.

Задача 9.8. Составьте инструкцию для вывода числа 100 в шестнадцатеричной системе

счисления с отображением основания системы счисления (0x). Для выполнения задания воспользуйтесь манипулятором `setiosflags()`.

Задача 9.9. В незавершенной программе имеется класс `strtype`. Для вывода строки на экран создайте пользовательскую функцию вывода:

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;
class strtype {
    char *p;
    int len;
public:
    strtype(char *ptr);
    ~strtype() { delete [] p; }
    friend ostream &operator <<(ostream &stream, strtype &obj);
};
strtype::strtype(char *ptr) {
    len = strlen(ptr)+1;
    p = new char(len);
    if (!p) {
        cout << "Allocation error\n";
        exit(1);
    }
    strcpy(p, ptr);
}
// Create operator << inserter function here.
int main() {
    strtype s1("This is a test."), s2("I hate C++!");
    cout << s1 << '\n' << s2;
    return 0;
}
```

Задача 9.10. Замените в следующей программе функцию `show()` пользовательской функцией вывода:

```
#include <iostream>
using namespace std;
class planet {
protected:
    double distance; // расстояние в милях от Солнца
    int revolve; // полный оборот в днях
public:
    planet(double d, int r) {
        distance = d;
        revolve = r;
    }
};
class earth : public planet {
    double circumference; // окружность орбиты
public:
    earth(double d, int r) : planet(d, r) {
        circumference = 2 * distance * 3.1416;
    }
    /* Rewrite this so that it displays the information using an inserter function.
*/
    void show() {
```

```

        cout << "Distance from sum: " << distance << '\n';
        cout << "Days in orbit: " << revolve << '\n';
        cout << "Circumference of orbit: " << circumference << '\n';
    }
};
int main() {
    earth obj(93000000, 365);
    cout << obj;
    return 0;
}

```

Задача 9.11. Добавьте пользовательскую функцию ввода в класс **strtype** Задачи 9.9.

Задача 9.12. Создайте класс для хранения целого и его наименьшего делителя. Создайте для этого класса пользовательские функции ввода и вывода.

Задача 9.13. Напишите программу для вывода числа 100 в десятичной, шестнадцатеричной и восьмеричной системе счисления. (Используйте флаги формата класса **ios**.)

Задача 9.14. Напишите программу для вывода значения 1000.5364 в 20-символьном поле, с выравниванием влево, с двумя позициями после запятой и символом * в качестве символа заполнения. (Используйте флаги формата класса **ios**.)

Задача 9.15. Перепишите ваши ответы на Задачи 9.13 и 9.14 так, чтобы использовались манипуляторы ввода/вывода.

Задача 9.16. Покажите, как записать и как отобразить для класса **cout** флаги формата. Используйте функции-члены либо манипуляторы.

Задача 9.17. Создайте для следующего класса пользовательские функции ввода и вывода:

```

class pwr {
    int base;
    int exponent;
    double result; // base to the exponent power
public:
    pwr(int b, int e);
};
pwr::pwr(int b, int e) {
    base = b;
    exponent = e;
    result = 1;
    for ( ; e; e--)
        result = result * base;
}

```

Задача 9.18. Создайте класс **box** для хранения размеров квадрата. Для вывода изображения квадрата на экран создайте пользовательскую функцию вывода. (Способ изображения выберите любой.)

Задача 9.19. Используя показанный здесь класс **stack**, создайте пользовательскую функцию вывода для вставки в поток содержимого стека. Покажите, что функция работает.

```
#include <iostream >
```

```

using namespace std;
#define SIZE 10
class stack {
    char stck[SIZE]; // holds the stack
    int tos; // index of top -of - stack
public:
    stack();
    void push(char ch); // push character on stack
    char pop(); // pop character from stack
};
// Initialize the stack:
stack::stack() {
    tos = 0;
}
// Push a character
void stack::push (char ch) {
    if (tos == SIZE) {
        cout << "Stack is full\n";
        return 0;
    }
    stck[tos] = ch;
    tos++;
}
// Pop a character:
char stack::pop() {
    if (tos==0) {
        cout << "Stack is empty\n";
        return 0;
    }
    tos--;
    return stck[tos];
}

```

Задача 9.20. Напишите программу с классом **watch**, который бы играл роль обычных часов. Используя стандартные функции времени, создайте конструктор класса, который должен считывать и запоминать системное время. Для вывода этого времени на экран создайте пользовательскую функцию вывода.

Задача 9.21. На основе класса, созданного для преобразования футов в сантиметры, напишите пользовательскую функцию ввода, формирующую строку-приглашение для записи числа футов. Кроме этого, напишите пользовательскую функцию вывода для отображения на экране как числа футов, так и числа сантиметров. Включите указанные функции в программу и продемонстрируйте их работоспособность.

```

class ft_to_cms {
    double feet;
    double cms;
public:
    void set(double f) {
        feet = f;
        cms = f * 30.48;
    }
};

```

Примечание: Фут (обозначается как «ft») — единица измерения длины. Фут равен 0,3048 м и используется в английской системе мер и США.