

Язык программирования Си++

Иванов А.П., Князева О.С.

Литература:

1. В.В. Подбельский. «Стандартный Си++» – Москва, Финансы и статистика, 2008, 688 с.
2. Б. Страуструп. «Язык программирования Си++. Специальное издание» – Москва, Бином, 2011, 1136 с.

Благодарности:

При подготовке данных методических материалов использованы семинары, разработанные Ермолаевым А.Г. и Козловым А.А. Авторы выражают им глубокую и искреннюю благодарность.

Семинар 1. Классы, конструкторы, деструктор, перегрузка операторов.

- Объектно-ориентированное программирование: инкапсуляция, наследование, полиморфизм.
- Классы как типы, объекты как переменные этих типов.
- Устройство класса «рациональное число» (числитель и знаменатель) с перегрузкой всех арифметических операторов, присваивания, конструкторами (инициализации и копирования), деструктором а так же - перегрузка операторов ввода-вывода.
- Оператор ::, объявление тела метода вне тела класса.
- Статические методы и поля.

```
#include <iostream>
using namespace std;

class Ratio {
private:
    int a,b;
public:
    explicit Ratio(int na=0, int nb=0) : a(na), b(nb)
    { cout << "Hi: (" << a << "/" << b << ")" << endl << flush; }
    Ratio(const Ratio& src) : a(src.a), b(src.b)
    { cout << "Hi: (" << a << "/" << b << ")" << endl << flush; }

    ~Ratio() { cout << "RIP: (" << a << "/" << b << ")" << endl << flush; }

    Ratio& operator=( const Ratio& src ) { a=src.a; b=src.b; return *this; }

    Ratio& operator+=( const Ratio& src )
    { a = a * src.b + b * src.a;
      b *= src.b;
      return *this;
    }
    int chislitel() const { return a; }
    int znamenatel() const { return b; }

    Ratio& reverse() { int tmp=a; a=b; b=tmp; return *this; }

    operator int() const { return a/b; }
```

```
Ratio operator+( const Ratio& src2 ) const
{ Ratio res(*this);
  res += src2;
  return res;
}
};

ostream& operator<<(ostream& os, const Ratio& src)
{ os << "(" << src.chislitel() << "/";
  os << src.znamenatel() << ")" << flush;
  return os;
}

istream& operator>> (istream& is,Ratio& r)
{
  int na, nb=1; is >> na;
  char c;      is >> c;

  if (c=='/') is>> nb;

  r = Ratio(na,nb);
  return is;
}

void main( )
{
  Ratio r1,r2,r3;
  char op;
  cout<<"enter r1"<<endl<<flush;
  cin>>r1;
  cout<<"enter op"<<endl<<flush;
  cin>>op;
  cout<<"enter r2"<<endl<<flush;
  cin>>r2;

  switch (op){
    case'+': r3=r1+r2; break;
    case'-': r3=r1-r2; break; /* этот и последующие арифметические */
    case'*': r3=r1*r2; break; /* операторы нужно реализовать */
    case'/': r3=r1/r2; break; /* в классе Ratio по образцу operator+ */
    default:
      cout<<"Error!"<<flush;
      return ;
  }

  cout<<r1<<' '<<op<<' '<<r2<<" = "<<r3<<flush;
}
}
```

Типовое задание: написать аналогичный класс и программу-калькулятор для него, выполняющую арифметические действия с объектами, вводимыми пользователем.

1. Вариант

Создать **класс комплексных чисел**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=». Перегрузить операции ввода и вывода. Написать программу–калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

2. Вариант

Создать **класс трехмерных векторов**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

3. Вариант

Создать **класс матриц 2×2**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор.

4. Вариант

Создать **класс времени** в формате чч:мм:сс.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор.

5. Вариант

Создать **класс даты** в формате дд.мм.гг.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

6. Вариант

Создать **класс треугольников**, катеты которых параллельны осям координат.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

7. Вариант

Создать **класс животных (волки и буйволы)**, живущих по определенным правилам.

Один волк может съесть одного буйвола, два буйвола могут затоптать волка. Поедание приводит к увеличению массы волка на массу буйвола, затаптывание – обнуляет массу волка. Два буйвола могут породить третьего, масса которого равна массе наибольшего из них. Два волка могут породить третьего, масса которого равна массе наименьшего из них. У созданных объектов должны быть определены конструкторы, оператор присваивания, все необходимые методы, операции ввода и вывода. Написать программу, которая иллюстрирует все имеющиеся методы по указаниям от пользователя.

8. Вариант

Создать **класс кругов**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

9. Вариант

Создать **класс прямоугольников**, со сторонами, параллельными осям координат.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

10. Вариант

Создать **класс полиномов 4 порядка**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции (+, -) с объектами данного класса.

11. Вариант

Создать **класс квадратных уравнений вида $ax^2+bx+c=0$** .

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции (+, -) с объектами данного класса.

12. Вариант

Создать **класс полиномов порядка N**.

N – произвольная, определяемая в программе константа. Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор.

13. Вариант

Создать **класс системных блоков** (частота процессора, количество памяти, стоимость).

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

14. Вариант

Создать **класс четырехмерных векторов**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «=» (умножение – скалярное произведение). Перегрузить операции ввода и вывода. Написать программу – калькулятор.

15. Вариант

Создать **класс полиномов 3 порядка**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Написать программу – калькулятор.

16. Вариант

Создать **класс рациональных чисел**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор.

17. Вариант

Создать **класс матриц 3×3**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор.

18. Вариант

Создать **класс воинских подразделений**: род войск, количество единиц (техники или людей), сила.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «=». Сложение и вычитание допустимо только в пределах одного рода войск. Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

19. Вариант

Создать **класс муравейников** (количество муравьев, занимаемая площадь муравейника, вид муравьев, сила муравейника).

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «=». Сложение и вычитание допустимо только для муравейников одного вида муравьев. Перегрузить операции ввода и вывода. Написать программу – калькулятор.

20. Вариант

Создать **класс двумерных векторов**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор.

21. Вариант

Создать **класс полиномов 5 порядка**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор.

22. Вариант

Создать **класс шаров**.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

23. Вариант

Создать **класс студенческих групп** (номер группы, количество студентов, фамилия старосты).

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «=». При арифметических операциях староста всегда сохраняется от левого операнда. Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.

24. Вариант

Создать **класс денежных сумм**, хранимых в формате `рубли.копейки`.

Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*», «/», «=» (умножение и деление – на вещественное число). Перегрузить операции ввода и вывода. Написать программу – калькулятор.

25. Вариант

Создать **класс матриц $N \times N$** .

N – произвольная, определяемая в программе константа. Определить в нем конструктор, деструктор, конструктор копирования. Перегрузить операции «+», «-», «+=», «-=», «*=», «=». Перегрузить операции ввода и вывода. Написать программу – калькулятор, который позволит пользователю производить арифметические операции с объектами данного класса.
