

Контрольное задание для студентов группы МПЗ-101 (средний уровень) (2 семестр)

Прислать решенное контрольное задание до 6 июня 2010 г. на krestnikova@gmail.com
Обращайтесь с вопросами по заданию по тому же e-mail.

Темы к прочтению для выполнения контрольного задания:

- Классы
 - Наследование
-

Задание №1.

Дано определение класса «Рациональные дроби»:

```
class Fraction
{
    public:
        Fraction(int Numerator = 1, int Denominator = 1);
        bool SetValue(int Numerator, int Denominator); /*установка
значения числителя и знаменателя, если знаменатель равен 0, то
метод возвращает false*/
        bool SetValue(const Fraction* fraction); /*установка
значения числителя и знаменателя из дроби, заданной в параметре,
если знаменатель равен 0, то метод возвращает false*/

        bool SetValue(const Fraction& fraction);
        void Add(const Fraction* fraction2); /*сложение с объектом
fraction2, результат присваивается текущей дроби*/

        Fraction Add(const Fraction& fraction2) const; /*сложение с
объектом fraction2, результат сложения возвращается*/

        Fraction Add(int Numerator, int Denominator) const;
/*сложение с объектом fraction2, результат сложения
возвращается*/

        void Subtract(int Numerator, int Denominator); /*вычитание из
текущей дроби дробь, числитель которой Numerator, а знаменатель
- Denominator, результат присваивается текущей дроби*/

        Fraction Subtract(const Fraction* fraction2)
const; /*результат возвращается*/
        void Subtract(const Fraction& fraction2);
        void Multiplication(const Fraction* fraction2); /* умножение
на объект fraction2*/
        Fraction Multiplication(const Fraction& fraction2) const;
        void Multiplication(int Numerator, int Denominator);
        void Print() const; //вывод на экран дроби
        void RaiseToDegree(int n); /*возведение в степень текущей
дроби*/
        void Divide(const Fraction* fraction2); /*делить текущую
дробь на объект, указатель на который хранится в fraction2*/
        void Divide(const Fraction& fraction2);
        void Inverse(); //обращение текущей дроби
```

```

void Cancel();//сокращение текущей дроби

Fraction& operator=(const Fraction&) const;
Fraction operator+(const Fraction&) const;
Fraction operator*(const Fraction&) const;
Fraction operator/(const Fraction&) const;
private:
    int numerator, denominator;
};

```

Реализовать и проверить работоспособность всех функций-членов.

Задание №2.

Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом.

Задание №3.

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве (или чуть посложнее, размер вектора известен на этапе выполнения).

Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, синуса угла между векторами. Написать программу, демонстрирующую работу с этим классом.

Задание №4.

Написать класс «Матрица», где поля класса (данные-члены класса) – элементы матрицы, операции с матрицами – методы класса (функции-члены класса).

Реализовать следующие операции с матрицами:

- Сложение, вычитание,
- умножение на число и матрицу,
- транспонирование,
- нахождение определителя матрицы,
- сравнение на равенство,
- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, верхняя треугольная, нижняя треугольная),
- вывод матрицы на экран.

Написать программу, демонстрирующую работу с этим классом.

Варианты: размеры матрицы фиксированы, размер матрицы определяются во время выполнения программы.

Задание №5.

Реализовать класс «Строка» (данные-члены класса – строка (указатель на массив, выделенной памяти на строку), функции-члены класса – операции, которые можно производить со строками, например, операция конкатенации присоединяет к одной строке

другую строку, при этом первая строка содержится в объекте типа «Строка», вторая строка содержится в объекте типа «Строка»).

```
String s1, s2, s3;//объявляет 2 объекта типа «Строка»  
s3 = s1.Concat(s2); /*присоединяем к строке s1 строку s2,  
результат присоединения помещается в s3, при этом строки s1 и s2  
не изменяются.* /
```

Реализовать следующие операции:

- Конкатенации
- Сравнения
- Преобразование в число (если возможно)

Проверить работоспособность класса.

Задание №6.

Создайте простую иерархию геометрических фигур, состоящую из базового класса Shape и производных классов Circle и Triangle.

Опишите в базовом классе поля и методы характерные для всех фигур одновременно. Включите в базовый класс виртуальную функцию draw() и переопределите её в производных классах (пусть к примеру результат выполнения это метода для классов Shape, Circle и Triangle будут соответственно вывод на экран следующих строк «draw::shape», «draw::circle» и «draw::triangle»).

Создайте массив указателей на объекты, созданные в куче, и вызовите draw() через указатели базового класса, чтобы проверить поведение виртуальных функций.
