Контрольная работа   
по дисциплине   
**«Объектно-ориентированное программирование»**

Каждое задание состоит в создании класса и решении с его помощью задачи.   
Кроме методов, перечисленных в задании, **если это имеет смысл**, должны быть реализованы:

* Конструктор по умолчанию;
* Конструктор, устанавливающий все свойства объекта (может совпадать с Конструктором по умолчанию);
* Копирующий конструктор;
* Деструктор;
* Методы для ввода объекта с консоли и вывода его на консоль;

Методы для получения и изменения свойств объекта.

Все поля должны быть скрытыми.

В соответствии с контекстом должны использоваться ключевое слово const и ссылки.

Там, где это имеет смысл, перегружать операторы. В списке методов приведен минимальный набор функций. В своей реализации следует придумать и реализовать еще несколько функций и операторов.

Например, класс комплексных чисел может иметь такое объявление:

classComplex {  
 <поля, методы требуемые по заданию>   
 Complex( doublere=0, doubleim=0 );   
 Complex( const Complex & );   
 Read();   
 Write() const;   
doubleGetIm() const;   
doubleGetIm() const;   
voidSet( doublere, doubleim=0 );  
 Complex operator + (const Complex & ) const;  
 …  
}

В тестовом примере, помимо основной задачи, добавить тестирование тех методов класса, которые заданием не затронуты.

# Дробь

Класс, предназначенный для хранения рациональных чисел в виде дроби (целые числитель и знаменатель). Дробь всегда должна быть несократимой.

## Методы

Инициализация дроби целым числом.

Получение целой части.

Получение обратной дроби.

## Операторы

* Присвоение целого числа.
* Преобразование в **double** .
* Арифметические операции с дробями (+,   –,   \*,   /).

Сравнение дробей (>, <, ==).

## Задача

Написать программу, которая вводит две дроби (A и B) и два целых числа (x и y), вычисляет значение выражения A2 + Bx + y и выводит его на экран в виде простой дроби и в виде десятичного числа.

# Дата (день, месяц, год)

Класс, предназначенный для хранения даты. Дата хранится в виде года, месяца и числа.

При вводе даты должен быть контроль корректности даты.

## Операторы

* Разность дат, как число дней между датами.
* Сравнение дат (>, <, ==).
* Сумма даты и числа (дней).

Инкремент и декремент даты.

## Задача

Программа должна ввести три даты –  D, D1 и D2.

Требуется найти дату D3, которая больше даты D2 на столько же дней, на сколько дата D2 больше даты D1.

Определить входит ли дата D в интервал [D1, D3].

# Дата (дни)

Класс даты, который реализован как количество дней, прошедших с 1 января 1990 года.

При вводе даты должен быть контроль корректности даты.

## Операторы

* Разность дат, как число дней между датами.
* Сравнение дат (>, <, ==).
* Сумма даты и числа (дней).

Инкремент и декремент даты.

## Задача

Программа должна ввести три даты –  D, D1 и D2.

Требуется найти дату D3, которая больше даты D2 настолько же дней, насколько дата D2 больше даты D1.

Определить входит ли дата D в интервал [D1, D3].

# Время (час, минута, секунда)

Класс для хранения времени суток. Время должно быть представлено в виде часов, минут и секунд.

При вводе должен быть предусмотрен контроль корректности.

При выходе времени за пределы суток должно происходить корректное преобразование, например, если к 23 часам прибавить 1 час 30 минут, должно получиться 0 часов 30 минут.

## Методы

Увеличение и уменьшение на час, и на минуту.

## Операторы

* Разность как количество секунд между двумя временами суток.
* Сумма времени и целого числа (секунд).
* Инкремент и декремент (по секунде).

Сравнение (>, <, ==).

## Задача

Программа должна ввести три значения времени –  T , T1 и T2.

Требуется найти время T3, которое позже времени T2 на столько секунд, на сколько время T2 позже времени T1.

Определить входит ли момент T во временной интервал [T1, T3].

# Время (секунды)

Класс для хранения времени суток. Время должно быть представлено в виде числа секунд, прошедших с начала суток.

При вводе должен быть предусмотрен контроль корректности.

При выходе времени за пределы суток должно происходить корректное преобразование, например, если к 23 часам прибавить 1 час 30 минут, должно получиться 0 часов 30 минут.

## Методы

Увеличение и уменьшение на час, и на минуту.

## Операторы

* Разность как количество секунд между двумя временами суток.
* Сумма времени и целого числа (секунд).
* Инкремент и декремент (по секунде).

Сравнение (>, <, ==).

## Задача

Программа должна ввести три времени – T, T1 и T2.

Требуется найти время T3, которое позже времени T2 на столько секунд, на сколько время T2 позже времени T1.

Определить входит ли момент T во временной интервал [T1, T3].

# Вектор

Класс для хранения векторов размерности N , где N – целая константа. Вектор хранится в виде массива координат.

## Методы

* Получение длины вектора.
* Проверка на ортогональность векторов.

Получение проекции на n -тую ось. (Результат – вектор).

## Операторы

* Сложение.
* Скалярное умножение.

Умножение на число.

## Задача

Для заданных векторов A , и B.

* Вычислить значение выражений:

.

Вывести число x, вектор C, а также его длину.

# Матрица

Класс для хранения матриц размерности 3. Матрица хранится в виде двухмерного массива.

## Методы

Получение определителя матрицы.

## Операторы

* Сложение.
* Умножение матриц.

Умножение на число.

## Задача

* Для заданных матриц A , и B вычислить значение выражения

.

* Вывести матрицу C, а также ее определитель.

# Многочлен

Класс многочленов степени не выше 10 с вещественными коэффициентами. Многочлен представляется в виде массива коэффициентов. Ввод должен быть организован так, чтобы не требовалось вводить старшие нулевые коэффициенты. Так, если вводится многочлен степени 2, требуется ввести только 3 коэффициента. При выводе старшие нулевые коэффициенты не должны выводиться.

## Методы

Получение степени многочлена.

## Операторы

* Сложение многочленов ( **+** ).
* Умножение многочленов ( **\*** ).

Значение в точке ( **()** ).

## Задача

Посчитать значение выражения в заданной точке x , где P – многочлен.

# Строка

Класс, реализующий строку в стиле языка Pascal .

Строка в языке Pascal, это массив символов, в котором нулевой байт содержит текущую длину сроки.

## Методы

* Получение длины строки.
* Получение подстроки.
* Удаление подстроки.

Поиск подстроки (получение индекса начала искомой подстроки.  
Если подстрока не найдена, результат равен –1).

## Операторы

* Сложение строк (+).
* Получение символа по индексу ([]).

## Задача

Вводится набор строк. Требуется в каждой строке вырезать слово “hello”, то что получится объединить в одну строку и вывести на экран.

# Множество символов

Класс для хранения множества символов. Внутреннее представление может быть массивом байтов (256 байт) или массива битов (32 байта). При этом i -тый байт (бит) содержит информацию о наличии во множестве символа с кодом i .

## Операторы

* Объединение множеств (+).
* Пересечение множеств (\*).
* Разность множеств (–)
* Проверка принадлежности символа множеству ([]).
* Сравнение множеств (==).

Проверка вхождения ( A < B , если A – подмножество B ).

## Задача

Даны две строки. Требуется вывести все символы, которые встречаются в первой строке, но не встречаются во второй.