

Вариант 1.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Комплексное число представляется парой действительных чисел (a, b) , где a — действительная часть, b — мнимая часть. Реализовать класс Complex для работы с комплексными числами. Обязательно должны присутствовать операции:

- сложения add, $(a, b) + (c, d) = (a + b, c + d)$;
- вычитания sub, $(a, b) - (c, d) = (a - b, c - d)$;
- умножения mul, $(a, b) * (c, d) = (ac - bd, ad + bc)$;
- деления div, $(a, b) / (c, d) = (ac + bd, bc - ad) / (c^2 + d^2)$;
- сравнение equ, $(a, b) = (c, d)$, если $(a = c)$ и $(b = d)$;
- сопряженное число conj, $conj(a, b) = (a, -b)$.

Реализовать методы: Init — инициализация, метод контролирует значения аргументов на корректность, Read — ввод с клавиатуры, Display — вывод на экран.

Вариант 2.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс vector3D, задаваемый тройкой координат. Обязательно должны быть реализованы: сложение и вычитание векторов, скалярное произведение векторов, умножение на скаляр, сравнение векторов, вычисление длины вектора, сравнение длины векторов. Реализовать методы: Init — инициализация, метод контролирует значения аргументов на корректность, Read — ввод с клавиатуры, Display — вывод на экран.

Вариант 3.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: для рублей и для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. Реализовать методы: Init — инициализация, метод контролирует значения аргументов на корректность, Read — ввод с клавиатуры, Display — вывод на экран.

Вариант 4.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Triangle для представления треугольника. Поля данных должны включать углы и стороны. Требуется реализовать операции: получения и изменения полей данных, вычисления площади, вычисления периметра, вычисления высот, а также определения вида треугольника (равносторонний, равнобедренный или прямоугольный). Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 5.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Angle для работы с углами на плоскости, задаваемыми величиной в градусах и минутах. Обязательно должны быть реализованы: перевод в радианы, приведение к диапазону 0-360, увеличение и уменьшение угла на заданную величину, получение синуса, сравнение углов. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 6.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Point для работы с точками на плоскости. Координаты точки — декартовы. Обязательно должны быть реализованы: перемещение точки по оси X, перемещение по оси Y, определение расстояния до начала координат, расстояния между двумя точками, преобразование в полярные координаты, сравнение на совпадение и несовпадение. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 7.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Рациональная (несократимая) дробь представляется парой целых чисел (a, b) , где a — числитель, b — знаменатель. Создать класс Rational для работы с рациональными дробями. Обязательно должны быть реализованы операции:

- сложения add, $(a, b) + (c, d) = (ad + bc, bd)$;
- вычитания sub, $(a, b) - (c, d) = (ad - bc, bd)$;
- умножения mul, $(a, b) * (c, d) = (ac, bd)$;
- деления div, $(a, b) / (c, d) = (ad, bc)$;
- сравнения equal, greater, less.

Должна быть реализована приватная функция сокращения дроби reduce (если 10/12, то функция reduce делает 5/6), которая обязательно вызывается при выполнении арифметических операций. Реализовать методы: Init — инициализация, метод контролирует значения аргументов на корректность, Read — ввод с клавиатуры, Display — вывод на экран.

Вариант 8.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Date для работы с датами в формате «год.месяц.день». Дата представляется тремя полями типа unsigned int: для года, месяца и дня. Класс должен включать не менее двух функций инициализации: числами и строкой вида «год.месяц.день» (например, «2004.08.31»). Обязательными операциями являются: вычисление даты через заданное количество дней, вычитание заданного количества дней из даты, определение високосности года, присвоение и получение отдельных частей (год, месяц, день), сравнение дат (равно, до, после), вычисление количества дней между датами. Реализовать методы: Init — инициализация, метод контролирует значения аргументов на корректность, Read — ввод с клавиатуры, Display — вывод на экран.

Вариант 9.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Time для работы со временем в формате «час : минута : секунда». Класс должен включать в себя не менее трех функций инициализации: числами, строкой (например, «23:59:59») и секундами. Обязательными операциями являются: вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, вычитание из времени заданного количества секунд, сравнение моментов времени, перевод в секунды, перевод в минуты (с округлением до целой минуты). Реализовать методы: Init — инициализация, метод контролирует значения аргументов на корректность, Read — ввод с клавиатуры, Display — вывод на экран.

Вариант 10.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Реализовать класс FuzzyNumber для работы с нечеткими числами, которые представляются тройками чисел $(x - e_l, x, x + e_2)$. Для чисел $A = (A - a_l, A, A + a_r)$ и $B = (B - b_l, B, B + b_r)$ арифметические операции выполняются по следующим формулам:

- сложение:

$$A + B = (A + B - a_l - b_l, A + B, A + B + a_r + b_r);$$

- вычитание:

$$A - B = (A - B - a_l - b_l, A - B, A - B + a_r + b_r);$$

- умножение:

$$A * B = (A * B - B * a_l - A * b_l + a_l * b_l, A * B, A * B + B * a_l + A * b_l + a_l * b_l);$$

- обратное число:

$$A = (1 / (A + a_r), 1 / A, 1 / (A - a_l)), A > 0;$$

- деление:

$$A / B = ((A - a_l) / (B + b_r), A / B, (A + a_r) / (B - b_l)), B > 0.$$

Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 11.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Реализовать класс Account, представляющий собой банковский счет. В классе должны быть четыре поля: фамилия владельца, номер счета, процент начисления и сумма в рублях. Открытие нового счета выполняется операцией инициализации. Необходимо выполнять следующие операции: сменить владельца счета, снять некоторую сумму денег со счета, положить деньги на счет, начислить проценты, перевести сумму в доллары, перевести сумму в евро, получить сумму прописью (преобразовать в числительное). Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 12.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Номиналы украинских гривен могут принимать значения 1, 2, 5, 10, 20, 50, 100, 500, 1000, 2000. Копейки представить как 0.01 (1 копейка), 0.02 (2 копейки), 0.05 (5 копеек), 0.1 (10 копеек), 0.25 (25 копеек), 0.5 (50 копеек). Создать класс Money для работы с денежными суммами. Сумма должна быть представлена полями-номиналами, значениями которых должно быть количество купюр данного достоинства. Реализовать сложение сумм, вычитание сумм, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. Дробная часть (копейки) при выводе на экран должны быть отделена от целой части запятой. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 13.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Реализовать класс Bankomat, моделирующий работу банкомата. В классе должны содержаться поля для хранения идентификационного номера банкомата, информации о текущей сумме денег, оставшейся в банкомате, минимальной и максимальной суммах, которые позволяет снять клиенту в один день. Сумма денег представляется полями-номиналами 10-1000. Реализовать метод инициализации банкомата, метод загрузки купюр в банкомат и метод снятия определенной суммы денег. Метод снятия денег должен выполнять проверку на корректность снимаемой суммы: она не должна быть меньше минимального значения и не должна превышать максимальное значение. Метод toString() должен преобразовать в строку сумму денег, оставшуюся в банкомате. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 14.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Fraction для работы с дробными числами. Число должно быть представлено двумя полями: целая часть — длинное целое со знаком, дробная часть — беззнаковое короткое целое. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 15.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Goods (товар). В классе должны быть представлены поля: наименование товара, дата оформления, цена товара, количество единиц товара, номер накладной, по которой товар поступил на склад. Реализовать методы изменения цены товара, изменения количества товара (увеличения и уменьшения), вычисления стоимости товара. Метод toString() должен выдавать в виде строки стоимость товара. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Вариант 16.

Создать класс с закрытыми полями, где операции реализуются как методы класса. Создать класс Payment (зарплата). В классе должны быть представлены поля: фамилия-имя-отчество, оклад, год поступления на работу, процент надбавки, подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начисленная и удержанная суммы. Реализовать методы: вычисления начисленной суммы, вычисления удержанной суммы, вычисления суммы, выдаваемой на руки, вычисления стажа. Стаж вычисляется как полное количество лет, прошедших от года поступления на работу, до текущего года. Начисления представляют собой сумму, начисленную за отработанные дни, и надбавки, то есть доли от первой суммы. Удержания представляют собой отчисления в пенсионный фонд (1 % от начисленной суммы) и подоходный налог. Подоходный налог составляет 13 % от начисленной суммы без отчислений в пенсионный фонд. Реализовать методы: Init – инициализация, метод контролирует значения аргументов на корректность, Read – ввод с клавиатуры, Display – вывод на экран.

Лабораторная работа №2. Объектно-ориентированное программирование .

Для задания лабораторной работы №3 реализовать:

- конструктор без аргументов;
- конструктор инициализации;
- конструктор копирования;
- конструктор с использованием списка инициализации;
- деструктор;
- константный объект класса;
- создать константные функции ввода-вывода и функцию get;
- осуществить отделение интерфейса от реализации;
- продемонстрировать работу указателя this.

Лабораторная работа №3 Объектно-ориентированное программирование .

Для задания лабораторной работы №3 все операции определить как внешние дружественные функции. Определить и продемонстрировать работу статической переменной и константной статической переменной.