

Лабораторная работа 15. Классы и объекты. Наследование

Часть I. Иерархия классов «персональные данные»

1. Создать класс Person, соответствующий личным данным человека:

- фамилия,
- имя,
- дата рождения (можно использовать time_t и структуру tm),
- пол (один символ)

Разместить все данные в закрытых полях, для доступа к ним создать геттеры и сеттеры..

При изменении данных о поле организовать проверку на корректность (ограничить допустимые значения символами 'm', 'f' или 'm', 'ж' ...)

Определить конструкторы, функцию консольного ввода в диалоге с пользователем, перегрузить операторы сравнения и операторы ввода и вывода <<, >>

Создать функцию-член класса, возвращающий возраст человека – количество полных лет.

Написать программу, вводящую данные об указанном пользователем количестве людей в динамический массив. Отсортировать массив по полю возраст в порядке возрастания; использовать собственную реализацию любого алгоритма сортировки.

Вывести для каждого элемента отсортированного массива фамилию, инициалы имени, возраст и пол

2. На основе класса Person сформировать производный класс «рабочий с фиксированным окладом», он должен содержать дополнительно поля:

- оклад (double),
- %премии (double)

Добавить и/или переопределить все необходимые функции.

Добавить функции, реализующие

- расчет общей суммы заработной платы с учетом премии
- расчет суммы подоходного налога 13%
- расчет суммы, выдаваемой на руки, за вычетом налога

Составить программу для тестирования этого класса

3. На основе класса «рабочий с фиксированным окладом» сформировать производный класс «рабочий с почасовой оплатой».

Добавить и/или переопределить все необходимые элементы такого класса самым минимальным образом:

- добавить в этот класс только одно поле – количество отработанных часов
- поле базового класса, понимаемое ранее как оклад, в новом классе должно хранить ставку почасовой оплаты
- соответственно переопределить только одну функцию из расчетных также можно переопределить функции ввода-вывода
- новые функции добавлять не надо

Составить программу для тестирования этого класса

Часть II. Иерархия классов «Геометрические фигуры»

1. Создать класс «точка на плоскости» (можно использовать реализацию из лекционного материала)

Класс должен включать

- поля для хранения координат точки на плоскости
- функции доступа к полям (геттеры, сеттеры)
- конструкторы
- функции или операторы для консольного ввода и вывода
- функцию расчета расстояния между двумя такими точками

2. Определить класс «цилиндр».

Считать, что окружность в основании цилиндра лежит на координатной плоскости $ХОУ$, известны координаты ее центра и какой-то точки на окружности.

Кроме того известна высота цилиндра

Класс должен включать поля (разместить в **protected**):

- точка на плоскости - центр окружности в основании цилиндра
- любая точка на окружности в основании цилиндра
- высота цилиндра (длина боковой поверхности)

Определить все необходимые функции (геттеры, сеттеры, конструкторы, ввода и вывода).

Добавить функции, рассчитывающие

- площадь основания
- длину окружности в основании
- объем цилиндра
- площадь боковой поверхности

Написать программу, проверяющую работу всех определенных функций.

3. На основе класса «цилиндр» определить класс «прямая треугольная призма»,

Класс должен включать поля

- три точки основания (три точки, с координатами (x_1, y_1) , (x_2, y_2) , (x_3, y_3) ,
две из них унаследованы от цилиндра, поэтому добавить надо только описание третьей точки)
- длина боковой стороны (унаследовано от цилиндра, добавлять не надо)

Определить (переопределить) все необходимые методы.

В секцию **protected** добавить вспомогательную функцию, рассчитывающую площадь треугольника по его сторонам.

Функции, рассчитывающие площадь основания и периметр основания должны быть переопределены

Функции расчета объема и площади боковой поверхности должны корректно работать без переопределения.

Написать программу, проверяющую работу определенных в классе функций.

4. На основе класса «прямая треугольная призма» определить класс «прямая призма с четырехугольным основанием», добавив координаты еще одной точки основания

Добавить/переопределить все необходимые поля и методы.

Добавить функции, определяющие

- является ли такая призма параллелепипедом
- является ли призма прямоугольными параллелепипедом
- является ли призма кубом

Написать программу, проверяющую работу функций класса.