

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ ООАиП

Задание 1.1	1
Задание 1.2	3

Язык программирования – C++

Задание 1.1

Реализовать иерархию классов в соответствии с вариантом. У всех вариантов в качестве базового класса выступает класс “Сотрудник”. Построить диаграмму классов.

В случае недопустимых значений полей выбрасываются исключения.

Примечание: если в задании требуется реализовать функционал для другого класса, которого нет в вашем варианте, делать этого не требуется

Варианты заданий:

- 1: 0 1 2 3
- 2: 0 1 2 4
- 3: 0 1 2 5
- 4: 0 1 2 6
- 5: 0 1 3 4
- 6: 0 1 3 5
- 7: 0 1 3 6
- 8: 0 1 4 5
- 9: 0 1 4 6
- 10: 0 1 5 6
- 11: 0 2 3 4
- 12: 0 2 3 5
- 13: 0 2 3 6
- 14: 0 2 4 5
- 15: 0 2 4 6
- 16: 0 2 5 6
- 17: 0 3 4 5
- 18: 0 3 4 6
- 19: 0 3 5 6
- 20: 0 4 5 6

0. Класс Директор:

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Также класс должен содержать поле, содержащее ФИО сотрудников, которые находятся в подчинении у директора, и их заработную плату.

Реализовать методы для увольнения и принятия работников.

Реализовать методы, позволяющие читать/писать из/в полей класса.

1. Класс Бухгалтер:

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Также класс должен содержать поле, содержащее ставку для всех должностей.

Реализовать метод для расчета заработной платы работникам, исходя из размера ставки.

Реализовать методы, позволяющие читать/писать из/в полей класса.

2. Класс Секретарь:

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Реализовать метод, который для данного директора выводит в виде таблицы список сотрудников.

Если в вашем варианте есть класс “Системный администратор”, то реализовать метод, который для данного системного администратора выводит в виде таблицы список оборудования.

Если в вашем варианте есть класс “Охранник”, то реализовать метод, который принимает массив охранников и выводит их в таблицу.

Если в вашем варианте есть класс “Водитель”, то реализовать метод, который принимает массив водителей и выводит их в таблицу.

Если в вашем варианте есть класс “Программист”, то реализовать метод, который принимает массив программистов и выводит их в таблицу.

Реализовать методы, позволяющие читать/писать из/в полей класса.

3. Класс Системный администратор:

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Класс должен содержать список оборудования, которым заведует системный администратор.

Реализовать методы, позволяющие читать/писать из/в полей класса.

4. Класс Охранник:

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Класс должен содержать поле, которое хранит в себе название спец. средства (дубинка, шокер и т.д.)

Реализовать методы, позволяющие читать/писать из/в полей класса.

5. Класс Водитель

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Класс должен содержать поле, которое хранит в себе массив из категорий прав.

Также должен содержать поле, содержащее массив транспортных средств, которыми управляет водитель.

Реализовать методы, позволяющие читать/писать из/в полей класса.

6. Класс Программист

Должен содержать имя, фамилию, отчество и зарплату. Данные поля должны находиться в закрытой области класса.

Класс должен содержать поле, которое хранит уровень программиста (junior, middle, senior, lead).

Класс должен содержать поле, которое будет содержать список технологий, которыми владеет программист.

Реализовать методы, позволяющие читать/писать из/в полей класса.

Задание 1.2

Реализовать дочерний класс на основе двух классов, реализованных в предыдущем задании 1.1.

Рассмотрите полученный результат и скажите, какие проблемы возникли при реализации данного множественного наследования (прим.: "алмаз смерти").

Создать новые классы "Инициалы" и "Выплаты", на основе которых реализовать дочерний класс "Данные о работнике".

Продемонстрировать работу класса.

В случае некорректных данных выбрасывать исключения.