**РАБОТА С КЛАССАМИ.**

На основе **вашего варианта из задания со словарями** вам необходимо разработать набор собственных классов.

Пример:

1. Необходимо написать программу, которая будет хранить информацию о сотрудниках. (например, такую как: номер, ФИО, дата рождения, оклад, находится сейчас в отпуске или нет).

Класс **Сотрудник (Employee)**, экземпляр которого инициализируется с аргументами, указанными выше.

Класс реализует (инкапсулирует) функциональность:

* **\_\_str(self)\_\_** — возвращает строковое представление в виде «Сотрудник 1 Иванов 28.10.2000, оклад: 30000, в отпуске: нет»**>**;
* **increase\_salary (summa)** — изменяет оклад;
* сотрудников можно сравнивать по дате рождения

Класс **Отдел (Department)**, экземпляр которого инициализируется с аргументами: название, список сотрудников, начальник.

Класс реализует (инкапсулирует) функциональность:

* **append(sotr)**— добавляет сотрудника в отдел;
* **\_\_str(self)\_\_** — возвращает строковое представление в виде «Отдел: Архив, начальник: Петров, Количество сотрудников: 4»;
* **print\_employees()** – печатает список сотрудников
* **print\_employees\_on\_leave()** - Вывод сотрудников отдела в отпуске/не в отпуске.

Приведите примеры работы с данными классами.

**См. пример данной классовой модели, файл: module\_employee.py**

Создайте схему ([mind-map](https://app.mindmup.com/)) с визуальным представлением структуры классов:

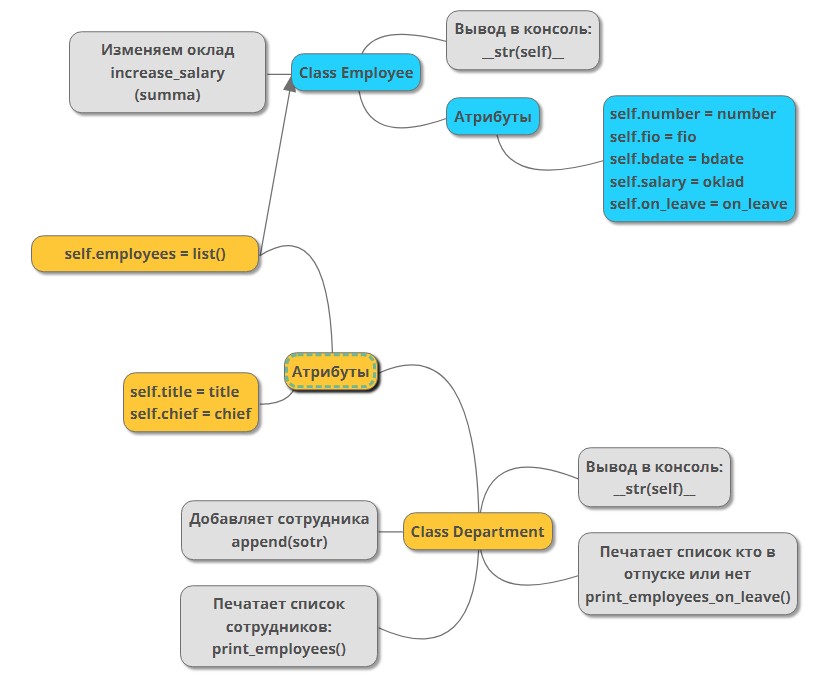


Рис.1 Классовая модель «Сотрудники»

**Варианты:**

**1**. Необходимо написать программу, которая будет хранить информацию о сотрудниках. (например, такую как: ФИО, пол, возраст, телефон, почта, должность, стаж).

Классы:

Сотрудник – содержит описанные выше атрибуты. Предусмотреть операции сравнения сотрудников по стажу.

Отделы - содержат сотрудников, имеют название, начальника, номер. Предусмотреть вывод сотрудников отдела; подсчет общего стажа всех сотрудников отдела.

Каждый сотрудник может иметь или не иметь детей, для этого создайте класс Дети (ФИО, возраст, пол)

Дополните модель собственными классами, свойствами и методами при необходимости.

**2.** Необходимо написать программу, которая будет хранить информацию о студентах. (например, такую как: ФИО, пол, возраст, телефон, почта, группа, курс).

Классы: Студент– содержит описанные выше атрибуты. Предусмотреть операции сравнения сотрудников по курсу.

Группы – содержат студентов; Атрибуты: название, номер. Методы: вывод всех студентов группы. Вывод всех мальчиков/ девочек группы. Предусмотреть операции сравнения по количеству студентов в группе.

Добавьте любой свой класс с методами.

Дополните модель собственными классами, свойствами и методами при необходимости.

**3.** Необходимо написать программу, которая будет хранить информацию о автомобилях автосалона. (например, такую как: госНомер, Марка, модель, пробег, год выпуска, кол-во владельцев, скидка, цена).

Классы:

Авто - содержит описанные выше атрибуты. Предусмотреть операции сравнения автомобилей по пробегу. Вывод информации о текущем автомобиле. (\_\_str()\_\_)

Владельцы – свойства: ФИО, паспорт, дата владения с, дата владения по, кол-во аварий на данном тс.

Класс Авто должен содержать данные о владельцах.

Аварии – дата, виновник (да/ нет), повреждения

Класс Авто должен содержать данные об авариях данного авто.

Предусмотреть вывод всей истории по указанному гос номеру авто.

Дополните модель собственными классами, свойствами и методами при необходимости.

**4.** Необходимо написать программу, которая будет хранить информацию о телефонных переговорах в словаре. (например, такую как: дата разговора, время разговора в минутах, город куда звонил, тариф за 1 мин.).

Классы:

Переговоры – содержит описанные выше атрибуты.

Клиенты – информация о клиентах. Атрибуты: ФИО, адрес, телефон, переговоры.

Предусмотреть операции клиентов по общему времени разговоров.

Класс Клиенты содержит список Переговоров.

Напишите вывод общего времени переговоров указанного клиента.

Дополните модель собственными классами, свойствами и методами при необходимости.

**5**. Необходимо написать программу, которая будет хранить информацию о путешествиях. (например, такую как: длительность поездки, стоимость, дата, страна).

Классы:

Туристы, свойства: ФИО, дата рождения, паспорт. Содержат информацию о поездках.

Поездки – содержит описанные выше атрибуты.

Необходимо сравнить поездки по их стоимости.

Написать метод выводящий общую длительность поездок указанного туриста.

Дополните модель собственными классами, свойствами и методами при необходимости.

class traveler:

def \_\_init\_\_(self, fio, bdate, passport):

self.fio = fio

self.bdate = bdate

self.passport = passport

class travel:

def \_\_init\_\_(self,duration,cost,date,country):

self.duration = duration

self.date = date

self.cost = cost

self.country = country

**6.** Необходимо написать программу, которая будет хранить информацию о рейсах авиакомпании. (например, такую как: Модель самолета, страна назначения. Страна вылета, время в пути (в минутах), дата вылета, кол-во проданных билетов).

Классы:

Рейсы – данные о рейсах компании. Атрибуты описан выше.

Пассажиры (ФИО, паспорт, телефон)

Предусмотреть процесс регистрации пассажира на рейс (добавить атрибут в класс Рейсы).

Реализовать возможность сравнения рейсов между собой по длительности в пути. Возможность вывода (метод) информации о рейсе и его пассажирах (по номеру рейса)

Дополните модель собственными классами, свойствами и методами при необходимости.

**7.** Необходимо написать программу, которая будет хранить информацию о товарах на складе. (например, такую как: Наименование, Цена. Ед. изм, кол-во).

Классы:

Товар – включает описанные выше атрибуты.

Склад – включает в себя товары и атрибуты: название, адрес, площадь.

Реализовать методы:

сравнения складов по площади;

вывод информации о всей продукции на указанном складе;

сравнение продукции по цене;

Метод пересчета количества продукции при смене единицы измерения. (Например, было 100 пакетов риса), в методе мы меняем единицу измерения на кг., а объем пакета указываем равным 900 гр, в итоге получим новое значение: 90 кг.

Дополните модель собственными классами, свойствами и методами при необходимости.

**8.** Необходимо написать программу, которая будет хранить информацию о оказанных услугах. (например, такую как: Наименование услуги, Стоимость. ФИО мастера, Дата оказания, Вид расчета).

Классы:

Услуги – содержат описанные выше атрибуты.

Клиенты – данные о клиентах. Свойства: ФИО, телефон, скидка, полученные услуги.

Реализуйте хранение услуг у клиентов.

Методы:

сравнение клиентов по размеру скидки;

сравнение услуг по стоимости;

вывод общей стоимости полученных услуг указанного клиента.

Создайте класс Мастер, придумайте для него необходимые свойства и методы.

Дополните модель собственными классами, свойствами и методами при необходимости.

**9.** Необходимо написать программу, которая будет хранить информацию о лекарствах. (например, такую как: Наименование, фирма, Кол-во, Цена, телефон).

Классы:

Аптека – атрибуты: номер, адрес, телефон.

Лекарство – атрибуты описаны выше.

Класс Аптека хранит данные о лекарствах.

Методы:

Сравнение лекарств по цене.

Вывод всех лекарств указанной аптеке.

Вывод самого дорого лекарства в указанной аптеке.

Дополните модель собственными классами, свойствами и методами при необходимости.

**10.** Необходимо написать программу, которая будет хранить информацию об авто в прокате. (например, такую как: Марка, модель, цена аренды в сутки, класс авто, цвет, пробег).

Классы:

Авто – атрибуты указаны выше.

Клиенты – атрибуты: ФИО, номер водит прав, телефон, скидка.

Аренда – содержит информацию об авто и его арендаторе, дату аренды с, по. Итоговую цену аренды.

Методы: сравнение клиентов по скидкам.

Авто, которое чаще всего берут в аренду.

Самого частого клиента.

Количество аренд авто на указанную дату.

Дополните модель собственными классами, свойствами и методами при необходимости.

**11.** Необходимо написать программу, которая будет хранить информацию о коммунальных платежах. (например, такую как: адрес, кол-во прописанных, площадь жилья, стоимость квартплаты).

Классы:

Недвижимость – атрибуты описаны выше.

Гражданин – ФИО, паспорт, телефон.

Класс Гражданин должен иметь свойство Недвижимость, для хранения объектов недвижимости.

Методы:

Сравнить граждан по количеству объектов недвижимости.

Сравнение недвижимости по стоимость квартплаты.

Метод, который будет пересчитывать стоимость квартплаты при изменении кол-ва прописанных и площади жилья.

Предусмотреть возможность продажи (или передачи) недвижимости от одного гражданина другому.

Дополните модель собственными классами, свойствами и методами при необходимости.

**12.** Необходимо написать программу, которая будет хранить информацию о видеопрокате. (например, такую как: Наименование диска, жанр, продолжительность в мин., стоимость в сутки).

Классы:

Диск – атрибуты указаны выше.

Клиенты: ФИО, паспорт, телефон.

Аренда: Атрибуты: Клиент, Диск, дата аренды с/ по, вернули или нет

Методы:

Сравнение дисков по продолжительности;

Вывод клиента и его истории видеопроката.

Клиентов, которые просрочили возврат.

Дополните модель собственными классами, свойствами и методами при необходимости.

**13.** Необходимо написать программу, которая будет хранить информацию о библиотеки. (например, такую как: Номер, Наименование книги, жанр, кол-во страниц, стоимость, автор, год издания).

Классы:

Книга – свойства описаны выше.

Читатель – ФИО, номер, телефон, адрес

Читательский билет – Книга, Читатель, дата когда взял, когда вернул.

Методы:

Сравнение книг по числу страниц;

Самого начитанного читателя;

Список задолжников, с тем учетом что срок возвраты книги не более 3-х недель;

Самую популярную книгу;

Дополните модель собственными классами, свойствами и методами при необходимости.

**14.** Необходимо написать программу, которая будет хранить информацию о абонентах сотового оператора. (например, такую как: название тарифа, абонентская плата, сколько Гб интернета, абонент).

Классы:

Абонент – свойства: фио клиента, паспорт, телефон, адрес.

Тариф – свойства описаны выше;

Методы:

Вывести всех абонентов указанного тарифа;

Сравнение тарифов по абонентской плате;

Вывести количество абонентов указанного тарифного плана;

Дополните модель собственными классами, свойствами и методами при необходимости.

**15.** Необходимо написать программу, которая будет хранить информацию о расписании поездов. (например такую как: название пункта назначения и отправления, номер поезда, время отправления, время в пути, станции на которых делается остановка (вложенный список)).

Классы:

Поезд – свойства описаны выше;

Пассажир – ФИО, паспорт, телефон;

Касса – Поезд, Пассажир, номер билета, цена;

Методы:

Вывести всех пассажиров указанного поезда;

Вывести общую стоимость проданных билетов по поездам;

Сравнить поезда по времени в пути;

Вывести поезд с самым большим числом остановок;

Вывести пассажира, чаще всего пользующимся услугами поездов и номер проданных ему билетов.

Дополните модель собственными классами, свойствами и методами при необходимости.

**16.** Необходимо написать программу, которая будет хранить информацию о книгах в домашней библиотеке (например, такую как: автор, жанр, год выпуска, кол-во страниц, издательство, название).

Классы:

Книга – свойства описаны выше.

Владелец – член семьи кто владеет книгой. Свойства: книга, имя владельца.

Методы:

сравнить членов семьи по количеству книг;

вывести список книг каждого члена семьи;

сравнить книги по кол-ву страниц;

вывести количество книг каждого жанра указанного члена семьи;

Дополните модель собственными классами, свойствами и методами при необходимости.

**17.** Необходимо написать программу, которая будет хранить информацию о фильмах в кинотеатре. (например, такую как: название, продолжительность, дата премьеры, режиссер, жанр, список главных актеров).

Классы:

Фильмы – свойства описаны выше.

Показ – фильм, дата показа, номер зала, цена билета;

Методы:

Сравнить фильмы по продолжительности;

Вывести список фильмов в диапазоне дат показа;

Вывести количество фильмов по залам показа;

Дополните модель собственными классами, свойствами и методами при необходимости.

**18.** Необходимо написать программу, которая будет хранить информацию о покупателях. (например, такую как: Фамилия, Имя, Отчество, Адрес, Дата рождения, Номер кредитной карточки и др).

Классы:

Покупатель – свойства описаны выше;

Товар – свойства: наименование, код, цена, описание.

Касса – свойства: Покупатель, Товар, кол-во товара, дата покупки, скидка.

Методы:

Возможность сравнивать товары по цене;

Сумму покупок указанного покупателя за весь период;

Сумму покупок за указанную дату;

Самый продаваемый товар;

Дополните модель собственными классами, свойствами и методами при необходимости.

**19.** Необходимо написать программу, которая будет хранить информацию об абонентах. (например, такую как: идентификационный номер, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Дебет, Кредит, Время междугородных и городских переговоров;).

Классы:

Абонент – свойства описаны выше.

Звонки – свойства: Абонент, город куда звонил, время разговора, дата разговора, стоимость минуты разговора;

Методы:

Внесение денег на счет абонентом (Дебет);

Списание денег со счета абонента (Кредит) в зависимости от стоимости минуты разговора;

Пересчет общего времени разговора абонента при совершении звонка;

Количество звонков указанного абонента;

Самый долгий разговор абонента;

Дополните модель собственными классами, свойствами и методами при необходимости.

**20.** Необходимо написать программу, которая будет хранить информацию о товарах на складе. (например, такую как: название товара, цена, количество, изготовитель, адрес производства, дата поступления товара, код товара).

Классы:

Товар – свойства описаны выше;

Склад – включает в себя товары и атрибуты: название, адрес, площадь.

Реализовать методы:

сравнения складов по площади;

вывод информации о всей продукции на указанном складе;

сравнение продукции по цене;

количество товаров на складе поступивших до указанной дате;

Дополните модель собственными классами, свойствами и методами при необходимости.

**21**. Необходимо написать программу, которая будет хранить информацию о рыбах в аквариуме в зоомагазине. (например, такую как: название, климат обитания, цена, продолжительность жизни, список чем питается, количество рыб и др).

Классы:

Рыба – свойства описаны выше;

Аквариум – свойства: объем, форма, дата очистки (можно в виде списка), рыбы.

Методы:

Сравнение рыб по цене;

Вывод всех обитателей указанного аквариума;

Продажа рыбки из аквариума (уменьшить кол-во или убрать совсем если кол-во стало = 0)

Сравнение аквариумов по количеству обитателей;

Дополните модель собственными классами, свойствами и методами при необходимости.

**22.** Необходимо написать программу, которая будет хранить информацию о питомцах приюта для собак. (например, такую как: кличка, возраст, вид животного, где был найден, вес, список прививок, есть ли опекун или нет, уникальный номер, примерная порода).

Классы:

Приют – свойства: Адрес приюта, кол-во собак, телефон, ФИО заведующего, питомцы;

Собака – свойства описаны выше;

Методы:

Сравнение собак по возрасту;

При добавлении/ удалении собаки в питомник пересчитывать общее количество собак в нём.

Вывести собак выбранного питомника по группам (есть или нет опекун)

Дополните модель собственными классами, свойствами и методами при необходимости.

**23.** Необходимо написать программу, которая будет хранить информацию о автомобилях автосервиса. (например, такую как: госномер, марка, список с оказанными услугами (можно вложенный список словарей)).

Классы:

Владельцы: - свойства: Фио, тел, скидка, список авто.

Авто: - свойства описаны выше;

Методы:

Сравнение авто по списку оказанных услуг (количество).

Пересчет скидки владельца в зависимости от кол-ва автомобилей у него.

Вывести данные владельца и все данные его автомобилей и оказанных им услуг;

Вывести услуги по конкретному авто.

Дополните модель собственными классами, свойствами и методами при необходимости.

**24.** Необходимо написать программу, которая будет хранить информацию о продажах в магазине. (например, такую как: название, код товара, цена, кол-во, скидка на товар, ед. измерения).

Классы:

Товары – свойства описаны выше;

Касса: Товар, дата продажи, список товаров, сумма.

Методы:

Сравнение товаров по цене;

Пересчет суммы продажи товаров при изменении списка товаров;

Количество товаров в операции по кассе;

Дополните модель собственными классами, свойствами и методами при необходимости.

**25.** Необходимо написать программу, которая будет хранить информацию об абонентах сотового оператора. (например, такую как: название тарифа, абонентская плата, сколько Гб интернета, есть ли пакет смс, есть ли пакет минут внутри сети, Абонент).

Классы:

Абонент: ФИО, паспорт, телефон

Тариф: - свойства описаны выше;

Методы:

Вывести данные об абонентах указанного тарифа

Сравнение тарифов по абонентской плате, если она одинаковая, то по кол-ву Гб интернета;

Вывести самый популярный тариф;

Перевод одного абонента с одного тарифа на другой;

Дополните модель собственными классами, свойствами и методами при необходимости.

**26.** Необходимо написать программу, которая будет хранить информацию о студентах. (например, такую как: ФИО, группа, возраст, список оценок по предметам, пол, специальность, номер зачетки).

Классы:

Группы: Номер, Название специальности; курс, Студент.

Студент – свойства смотри выше;

Методы: сравнение студентов по возрасту;

Вывод студентов указанной группы;

Перевод студента из одной группы в другую;

Оценки студента по предметам;

Число отличников в выбранной группе (отличник – когда все 5 по всем предметам)

Дополните модель собственными классами, свойствами и методами при необходимости.

**27**. Необходимо написать программу, которая будет хранить информацию о коммунальных платежах. (например, такую как: адрес, кол-во прописанных, площадь жилья, стоимость квартплаты, уникальный номер).

Классы:

Собственник: свойства: ФИО, паспорт, телефон, Недвижимость, Итоговая стоимость квартплаты

Недвижимость – свойства описаны выше;

Методы:

Сравнение недвижимости по площади;

Вывод объектов недвижимости Собственника;

Пересчет итоговой стоимости квартплаты при манипуляциях с недвижимостью;

Вывод количества объектов недвижимости по каждому собственнику.

Дополните модель собственными классами, свойствами и методами при необходимости.