**Создать класс Person содержащий:**

Поля: фамилия, пол, возраст.

(тут реализация как знаете: через конструктор инициализация полей, можете использовать свойства)

* Реализована возможность выводить информацию об объекте с помощью стандартного вызова print()

**Создать абстрактный класс-наследник Student:**

Все поля наследуются из родительского класса. Предусмотреть значение полей по умолчанию.

+

* Содержит поле: количество посещенных занятий **attended\_classes** .
* Расширить метод вывода информации об объекте добавив количество посещенных пар
* Абстрактный метод Test – будет возвращать строку: сдал или не сдал зачет.
* Реализовать подсчет созданных экземпляров класса (как будто проверяем сколько зарегистрировалось студентов) (PS хочу увидеть знания статических методов)

На его (**Student)** основе реализовать дочерние классы:

* Ordinary - обычный студент,
* Clever - умный студент,
* Genius - гениальный студент.

**Каждый дочерний класс содержит:**

переопределить метод Test, который возвращает строку "Best congratulations for passing exam", если студент сдал экзамен, и "Sory, you shall not pass" – если не сдал:

* + Обычный студент сдает зачет, если посетил 20 занятий.
  + Умный студент сдает зачет, если посетил больше 10 занятий
  + Гениальный студент сдает зачет, если посетил хотя бы одно занятие.

Продемонстрировать работу вашего класса. Будет плюсом, если сохраните в виде модуля, который работает в двух режимах: как подключаемый модуль, если нет, то реализован тестовый прогон.

**Пример работы с программой:**

