# Инвариантная самостоятельная работа №3

**Задание 3.1-3.2**

**Постановка задачи:**

3.1. Разработка классов и объектов «запись», «комментарий» для приложения «Блог» (использование наследования).

3.2. Создание геттеров и сеттеров для классов «запись», «комментарий» приложения «Гостевая книга». Создание функций для вывода на печать информации, хранящийся в объектах.

**Решение:**

[**https://repl.it/@AnastasiaTs/Isr31**](https://repl.it/@AnastasiaTs/Isr31)

**Код программы:**

**class Post():**

**"""**

**Функция создания поста в гостевой книге**

**(с) Царулкова Анастасия 2 группа 2020**

**"""**

**def \_\_init\_\_(self,nick='Nick',text='Hello world!'): # Инициализация экземпляра класса**

**self.\_\_verify\_text(nick) # Проверка псевдонима**

**self.\_\_verify\_text(text) # Проверка текста**

**self.\_\_user\_nick = nick # Присвоение значения свойству экземпляра**

**self.\_\_user\_text = text # Присвоение значения свойству экземпляра**

**"""Ник"""**

**@property**

**def nickname(self):**

**return self.\_\_user\_nick # getter**

**@nickname.setter**

**def nickname(self, nick): #setter**

**try:**

**self.\_\_verify\_text(nick)**

**except ValueError as e:**

**print(e)**

**else:**

**self.\_\_user\_nick = nick**

**"""Текст записи"""**

**@property**

**def textPost(self):**

**return self.\_\_user\_text**

**@textPost.setter**

**def textPost(self, text):**

**try:**

**self.\_\_verify\_text(text)**

**except ValueError as e:**

**print(e)**

**else:**

**self.\_\_user\_text = text**

**def \_\_verify\_text(self, text,field='text'):**

**if len(text) == 0:**

**raise ValueError(f'Поле {field} обязательное поле')**

**def \_\_str\_\_(self): #Вывод содержания экземпляра**

**return f"Author:{self.nickname} Text:{self.textPost}"**

**class Comment(Post): # Наследует класс Post**

**"""**

**Функция создания комментария под постом в гостевой книге**

**(с) Царулкова Анастасия 2 группа 2020**

**"""**

**def \_\_init\_\_(self,nick, text, rate=0):**

**Post.\_\_init\_\_(self, nick, text) # вызов инициализатора класса, от которого наследует Comment**

**self.\_\_rate = rate**

**@property**

**def comRate(self):**

**return self.\_\_rate**

**@comRate.setter**

**def comRate(self,rate):**

**self.\_\_rate = rate**

**def addRate(self): #Метод класса, который увеличивает рейтинг**

**self.\_\_rate += 1**

**def \_\_str\_\_(self): #Вывод содержания экземпляра**

**return f"Author:{self.nickname} Text:{self.textPost} Rating:{self.\_\_rate}"**

**Результат работы:**

**re = Post('TwilightSparkle','Dear princess Celestia...')**

**print(re.nickname)**

**print(re.textPost)**

**print(re)**

**n = Comment('Nana','Hi everyone!')**

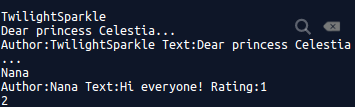
**print(n.nickname)**

**n.addRate()**

**print(n)**

**n.addRate()**

**print(n.comRate)**

****