FERNANDO TAMAYO GRADOS
December 9, 2023
IT FDN 110 B
Assignment 7
https://github.com/fernandotg123/IntroToProg-Python-Mod07

Assignment 7

1. Intro

This assignment doubles down on functions and classes in Python, expanding on the implementation of classes. In particular, it focuses on the data classes, which allows to manage data about people and students, through attributes i.e., characteristics, and constructors.

2. Data Classes

The program now has three types of classes: data classes, processing classes and presentation classes. In the previous assignment we worked on the latter two classes. To enhance the program, there will be two classes: Person, which will hold first and last name, including all attributes. And then, Student (Person), which will describe students and the course they are taking. For a student to exist, a person must exist first.

3. Class Person

See the code below to identify all attributes of the “person”:

```
class Person:
    """
    A class representing person data.

    Properties:
    - first_name (str): The person's first name.
    - last_name (str): The person's last name.

    ChangeLog:
    - Fernando Tamayo Grados, 12/9/2023, Executed Homework
    """
    first_name: str = ''
    last_name: str = ''

    def __init__(self, first_name: str = "", last_name: str = ""):
```

```
        self.first_name = first_name
        self.last_name = last_name

    @property
    def first_name(self):
        return self.__first_name.title()

    @first_name.setter
    def first_name(self, value: str):
        if value.isalpha() or value == "":
            self.__first_name = value
        else:
            raise ValueError("The first name should not contain numbers.")

    @property
    def last_name(self):
        return self.__last_name.title()

    @last_name.setter
    def last_name(self, value: str):
        if value.isalpha() or value == "":
            self.__last_name = value
        else:
            raise ValueError("The last name should not contain numbers.")

    def __str__(self):
        return f"{self.first_name},{self.last_name}"
```

4. Class Student (Person)

Also, here is the code for students:

```
class Student(Person):
    """
        A class representing student data.

        Properties:
        - first_name (str): The student's first name.
        - last_name (str): The student's last name.
        - course (str): The name of the course.

        ChangeLog:
        - Fernando Tamayo Grados, 12/9/2023, Executed Homework.
    """
    def __init__(self, first_name: str = "", last_name: str = "",
course_name: str = ""):
        super().__init__(first_name=first_name, last_name=last_name)
        self.course_name = course_name
```

8. Conclusion

Overall, the rest of the classes works as expected, thus running the program without issues.