Sid Tolbert
12/02/2024
IT FDN 110 A
Assignment07
https://github.com/Sid-Tolbert/IntroToProg-Python-Mod07

Assignment 7 – Classes and Objects

Introduction

Assignment 07 was about learning how to use object-oriented programming (OOP) principles to create organized code. The module introduced concepts like classes, constructors, properties, and inheritance, along with encapsulation to manage data. In addition, I explored using Git and GitHub Desktop for version control, which provided a structured way to manage and share code. These tools and principles were great for writing programs.

Doing the Assignment

The module materials provided a foundation for understanding how classes can be used to group data and functionality. For the assignment, I created a course registration program that demonstrated these principles in practice.

The program started with a Person class to handle general attributes like a first and last name.

```python
1 usage
class Person:
    """
    Creates a class called Person, a parent class

    ChangeLog: (Who, When, What)
    Sid Tolbert,12.1.2024,Created Class
    """
    def __init__(self, first_name:str, last_name:str):
        self.first_name = first_name
        self.last_name = last_name

    2 usages
    @property
    def first_name(self)->str:
        '''
        Returns the first_name as a title
        :return: the first name, properly formatted
        '''
        return self._first_name.title()

    1 usage
    @first_name.setter
    def first_name(self,value:str) -> None:
        '''
        Sets the first name, while doing validations
        :param value: The value to set
        '''
        if value.isalpha():
            self._first_name=value
        else:
            raise ValueError("First name must be alphabetic.")
```

Figure 1: Person Class

From there, I extended it into a Student class to include course-related details. This use of inheritance allowed me to build on the functionality of the Person class while adding new features specific to students.

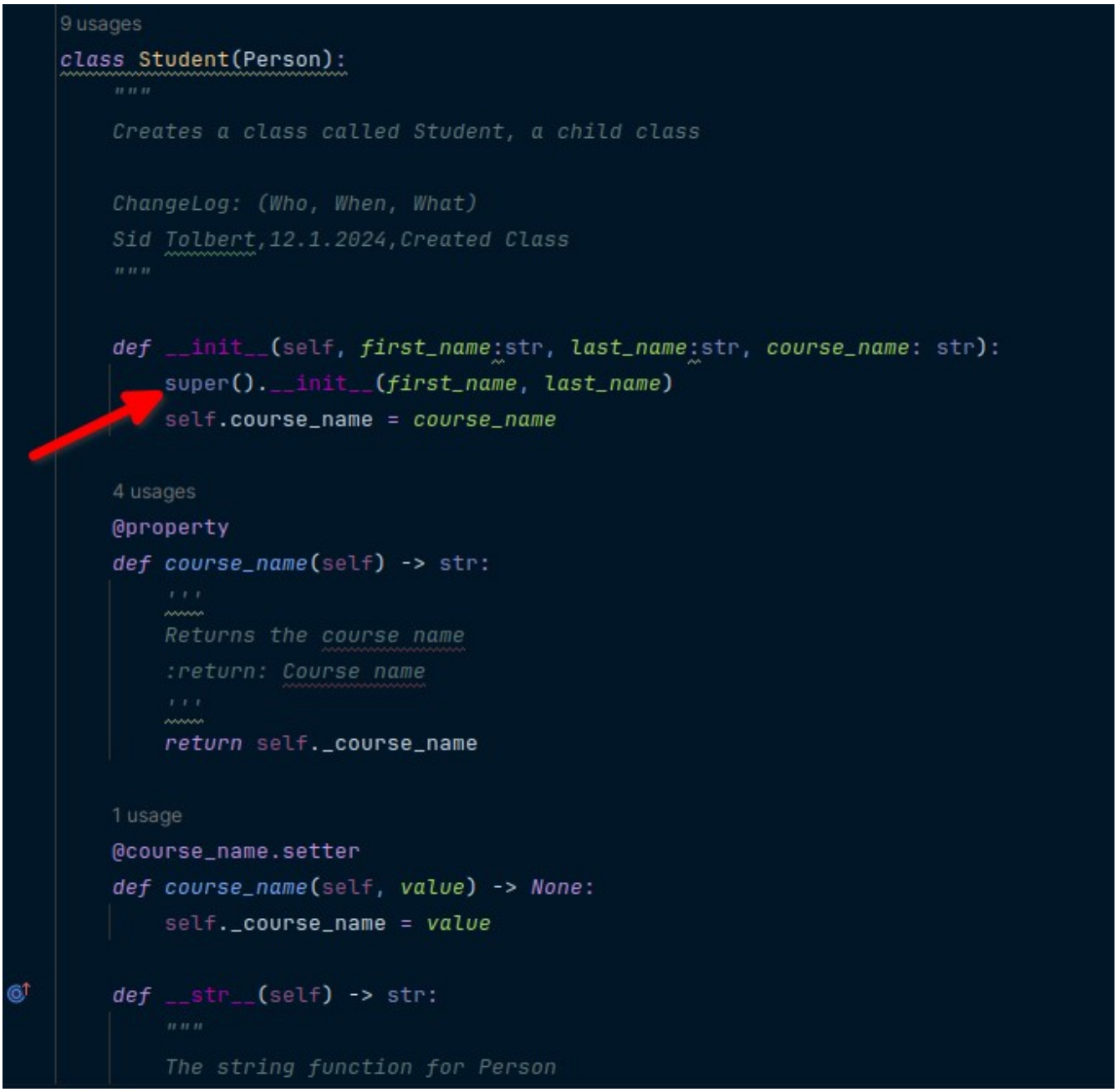

Figure 2: Student Class - Inheritance

Encapsulation was implemented through properties, which ensured data validation and control. For example, the program checked that names contained only alphabetic characters and rejected invalid input.

The program also incorporated file handling, making it possible to save and retrieve registrations. This ensured that user data persisted across sessions. Error handling was included to catch invalid inputs, such as entering non-alphabetic characters for student_name, and to provide meaningful feedback without causing the program to crash.

```python
def write_data_to_file(file_name: str, student_data: list[Student]):
    """ This function writes data to a json file with data from a list of dictionary rows

    ChangeLog: (Who, When, What)
    Sid Tolbert,12.1.2024,Created function

    :param file_name: string data with name of file to write to
    :param student_data: list of dictionary rows to be writen to the file

    :return: None
    """
    file_data = []
    for student in student_data:
        file_data.append({'first_name': student.first_name,
                          'last_name': student.last_name,
                          'course_name': student.course_name})
    file = None
    try:
        file = open(file_name, "w")
        json.dump(file_data, file)
        file.close()
        IO.output_student_and_course_names(student_data=student_data)
    except Exception as e:
        message = "Error: There was a problem with writing to the file.\n"
        message += "Please check that the file is not open by another program."
```

Figure 3: File Processing - adding in file_data

Using Git and GitHub Desktop for version control allowed me to track my progress and make changes confidently. Committing my work regularly made it easier to manage the development process, while uploading to GitHub provided a central place to store and share the final files.

Conclusion

This assignment highlighted how object-oriented programming can simplify and enhance code organization. Using inheritance made it easy to reuse and extend functionality, while encapsulation ensured data integrity and reduced the likelihood of errors. The practical experience with Git and

GitHub Desktop reinforced the importance of version control in managing code efficiently. These skills and concepts are invaluable for creating well-structured and scalable programs.