System Analysis Project 2022/23

OOP Encapsulation

By:

RAZVAN GOREA & FILIP BUMBU

Contact Info:

razvan.gorea2@mail.dcu.ie

filip.bumbu2@mail.dcu.ie

INTRODUCTION

Imagine an online banking system,
changes are constantly being made in regard to users' online banking accounts.

How can the bank and its online system ensure data protection and security to its users accounts, so only the users themselves can modify their accounts?

This is where oop encapsulation comes into play. . .

Points

Encapsulation

01 Data Hiding

02 Flexibility

03 Reusability

01 Data Hiding

Main Purpose

Used **to hide** the data implementation details while **exposing** the data abstraction to the outside world.

Protected Members

Members of a class which can only be accessed **within** the class or its subclasses.

Private Members

Members of a class which can **only** be accessed from the class.

02 Flexibility

Accessible

Setting the members **read-only** or **write-only**, depending on the **developer's needs**.

Testing

It helps in **unit testing**, which can be automated and thus **restricting** direct access to certain data would be useful.

Reliability

The Object-oriented system promises to be far more reliable than traditional systems, encapsulation being one of the system's concepts, **commonly seen** and **widely used**.

03 Reusability

Development

Encapsulation makes it easier to **maintain** software. It **speeds up** application development and **improves** the final product's quality.

Time

It helps us **reduce** the complexity of our design, by **reducing** the tight coupling between different components in our code.

Maintenance

Encapsulation **contributes** to a loose coupling which in turn leads to more **maintainable** code.

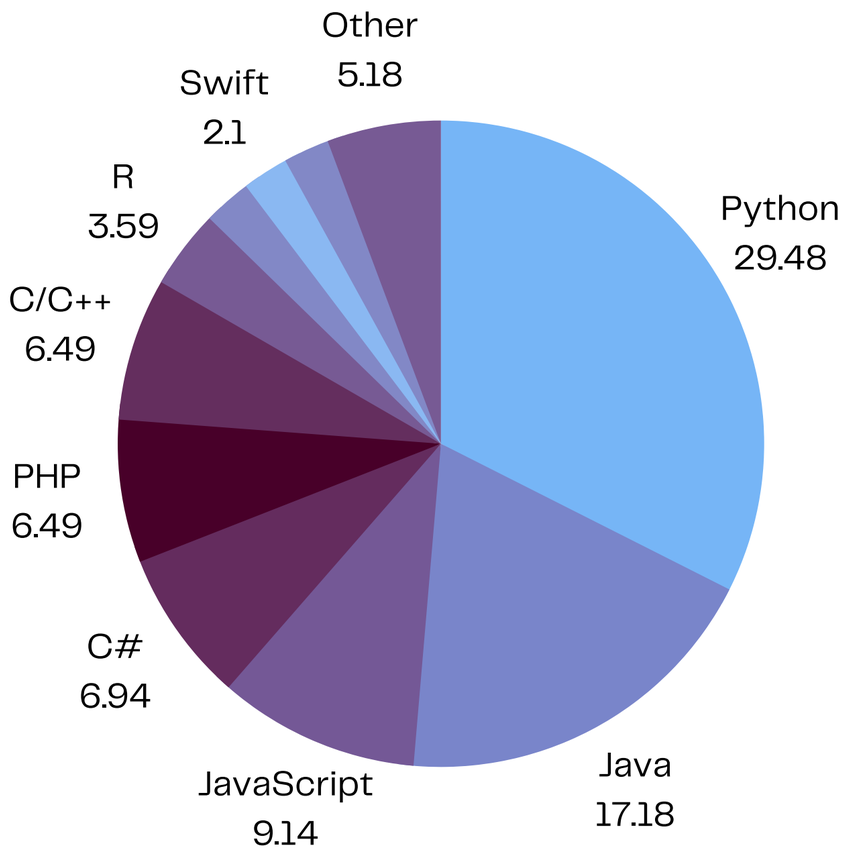
Code

Implementing Encapsulation in Python 3.10

Tip: read the comments in the code to better understand it.

```
class Base:
    def __init__(self):

        #Protected Number
        self._a = 2

# the protected members are denoted by the single "_" at the start

class Derived(Base):
    def __init__(self):

        # calling the protected number
        Base.__init__(self)
        #print(self._a)

        # modifying the protected number
        self._a = 3

obj_1 = Derived()
obj_2 = Base()

print(obj_1._a)
print(obj_2._a)
--------------------------------------------------------------------------------
class Base2:
    def __init__(self):

        # Private number
        self._a = 1
        self.__b = 2

class Derived2(Base2):

    def __init__(self):

        # calling the protected number and the private number
        Base2.__init__(self)
        print(self._a)
        print(self.__b)

obj_3 = Base()

print(obj_3._a)
print(obj_3.b)
```

Convention



Due to the reliability of OOP Languages, they are widely used, as you can see from the given chart.
The chart presents the popularity of programming languages based on market share.

References

- A. Qais, “Database, SQL and NoSQL,” Medium, 14-Jan-2021. [Online]. Available: https://medium.com/analytics-vidhya/database-sql-and-nosql-16865f90d634. [Accessed: 07-Oct-2022].
- “What's encapsulation, and what are the benefits of using encapsulation?,” Quora. [Online]. Available: https://www.quora.com/Whats-encapsulation-and-what-are-the-benefits-of-using-encapsulation. [Accessed: 07-Oct-2022].
- “C++ Access Specifiers,” C++ access specifiers. [Online]. Available: https://www.w3schools.com/cpp/cpp_access_specifiers.asp#:~:text=private%20%2D%20members%20cannot%20be%20accessed,learn%20more%20about%20Inheritance%20later. [Accessed: 07-Oct-2022].
- D. Braunschweig, “Encapsulation,” Programming Fundamentals, 15-Dec-2018. [Online]. Available: https://press.rebus.community/programmingfundamentals/chapter/encapsulation/#:~:text=Encapsulation%20is%20used%20to%20hide,parties'%20direct%20access%20to%20them. [Accessed: 07-Oct-2022].
- “Encapsulation (computer programming),” Wikipedia, 04-Sep-2022. [Online]. Available: https://en.wikipedia.org/wiki/Encapsulation_(computer_programming)#:~:text=Encapsulation%20is%20used%20to%20hide,is%20not%20unique%20to%20OOP. [Accessed: 07-Oct-2022].
- “Encapsulation in C: Working & need of encapsulation in C with examples,” EDUCBA, 03-Aug-2022. [Online]. Available: https://www.educba.com/encapsulation-in-c/. [Accessed: 07-Oct-2022].

References

- “Encapsulation in Java,” GeeksforGeeks, 03-Jun-2022. [Online]. Available: https://www.geeksforgeeks.org/encapsulation-in-java/. [Accessed: 07-Oct-2022].

- T. T. Contributor, “What is unit testing? definition from whatis.com,” SearchSoftwareQuality, 15-Aug-2019. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/unit-testing. [Accessed: 07-Oct-2022].

- “Advantage of OOP Languages,” Advantages and disadvantages of object-oriented approach. [Online]. Available: http://www.dba-oracle.com/t_object_oriented_approach.htm#:~:text=Improved%20Reliability%20and%20Flexibility%3A%20Object,be%20created%20at%20any%20time. [Accessed: 07-Oct-2022].

- “Development of Encapsulated Class Complexity Metric,” ScienceDirect.com. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017312004021/pdf?md5=c778fc47e54d8092fe550bb3e96edd90&pid=1-s2.0-S2212017312004021-main.pdf. [Accessed: 07-Oct-2022].

- D. Bolton, “What is encapsulation in C++ and C#?,” ThoughtCo, 10-Feb-2019. [Online]. Available: https://www.thoughtco.com/definition-of-encapsulation-958068. [Accessed: 07-Oct-2022].

LINK TO VIDEO

https://drive.google.com/file/d/1AubWg1hVca8lSVJczD5yF9vegRvlG-8P/view?usp=sharing

LINK TO SLIDES

LOOK INTO THE README.txt PROVIDED