ISSN 2278-3091

Volume 10, No.2, March - April 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse401022021.pdf

https://doi.org/10.30534/ijatcse/2021/401022021

A Research on Object Oriented Programming and Its Concepts

Rajesh Babu Nagineni[1]

[1]Assistant Professor, Dept. Of Computer Science and Engineering, KLEF, Andhra Pradesh, India

ABSTRACT

Object Oriented Software Engineering(OOSE) is popular and widely used in developing the real time complex projects and products due to innovations in the Software Engineering and the Development of Software Industry. Object Oriented Programming(OOP) is one of the key elements of Object Oriented Software Engineering(OOSE) besides Object Oriented Analysis(OOA) and Object Oriented Design(OOD).Object Oriented Programming(OOP) is one of the Programming Paradigms for the past few decades. The OOP Paradigm Has oven taken the Procedure Oriented Paradigm or POP, where real time complex projects an products can be developed by using OOP in an easy and reliable way. The main difference between the Object Oriented Programming(OOP) and Procedure Oriented Programming(POP) is "Data controlling access to Code" in OOP where as "Code acting on Data" in POP. In this Paper I am going to discuss about the Principles of Object Oriented Programming(OOP) , its concepts and some of the Popular and widely used Object Oriented Programming Languages. I also discussed the difference between Object Oriented Programming and Procedure Oriented Programming. Merits and Issues of Object Object Oriented Programming

Key words: Object Oriented Programming, Software Development, Object Oriented Programming Languages, Object Oriented Programming Paradigm, Object Oriented Programming Features

1.INTRODUCTION

Object Oriented Programming Languages(OOPLs) implements some or all the Object Oriented Programming(OOP) principles. The first and foremost Object Oriented Programming language is Simula, Later on developed into a full fledged OOP Language named as Simula 67 in the 1960's.SmallTalk,popularly know as a fully OOP Language developed on the basis of Simula 67 in 1970's. C++ is also a Object Oriented Programming Language invented in the late 1980's.In the 1990's Object Oriented Programming Languages such as Java,C#, Ruby has over taken the Procedural Oriented Programming Languages and invaded the Software Industry by their simplicity and reliability of Developing real time complex software projects.OOP languages mainly use the feature of Binding an hiding of code(functions) and data(variables) into a single unit called as "Class". The OOP Languages are designed and developed to meet the requirements of real world Problems. Object Oriented Programming Paradigm defines some principles like Encapsulation, Inheritance and Polymorphism which were implemented by OOP Languages. The OOP Languages implements higher level of abstraction using the concept of "Class". The OOP is widely used in designing and implementing Complex Projects ranging from Client/Server to Real Time. In Contrast Procedural Oriented Programming(POP) Languages are not best fit for real world problems as they lack te Reliability and Reusability features of Software. The POP Systems mostly fail due to the issues in design phase and Development of software.

2.OBJECT ORIENTED PROGRAMMING PARADIGM

Object Oriented Programming Paradigm is the basic technique of programming of Object Oriented Programming Languages.OOP Paradigm's main strength is "Abstraction" ,the others are "Encapsulation", "Inheritance" and "Polymorphism". All the OOP Languages more or less implements these features of OOP. The Programs of OOP contains objects(data) and a set of relational classes which facilitates the reusability(Inheritance) and flexibility(Polymorphism) of code. If a big and complex project is divided Into a set of relational classes, then it is possible to achieve the low production time and cost ,their by Complexity of the project operations is reduced and real time big and complex projects can be managed In an efficient way.OOP facilitates the users/developers to hide the complexity of complex Software in The concept called as "Classes". Object Oriented entire code of a project is based on the "Objects", The OOP pave the Programming(OOP) has its orientation on "Objects" That means the way to model the Classes and Objects. All of the objects in a project used to send and receive the messages, this is one of The important concepts of OOP ,called as "Message Passing". Programs of OOP makes use of Data and Code, which provides reliability and reusability of code, there by big and complex projects can be developed fast.

OBJECT ORIENTED PROGRAMMING FEATURES

Object Oriented Programming System(OOPS) defines the following features. All the modern and old OOP languages might implement some or all these features

(a) Classes

A Class forms basis of any OOP Program. Class contains Code(functions) and Data(variables) in it Classes abstracts the code and data from outside interference and misuse by the unauthorized Persons. class is a logical programming construct, that means it is imaginary and does not have existence in the real world. Every class of a OOP Program contains few distinct Objects which are Used for "Message

Passing”. Message Passing means manipulation of functions and data by the Objects, In other words objects can pass and get messages from code(functions).Message Passing Is one of the very important concepts of OOP .class also allow the other classes to derive its attributes(variables) and code(functions),This pave the way for Hierarchical Abstractions, These Abstractions specify which class/classes should execute first. Class is the basic element to implement the Concepts of OOP

(b) Objects

Objects are the key element of any OOP Program. Objects contains state(attributes) and behaviour or data of it and behaviour describes the Functionality.An Object is an instance of a Class. Unlike classes objects are having existence in the real world,in other words objects are physical entities. For Example an employee might contains attributes as empno,name,dept and salary etc. And behaviour as he/she might work and take leave etc. In this context employee is an object which consists of some attributes and behaviour. objects Might contains private data that may be only accessed by the some authorized people ,this data should not be accessed by unauthorized people. this restriction on data access can be achieved by access specifiers mechanism of many of the OOP Language .the other important aspect of objects are “Message Passing”.It means the objects can send and receive messages
From functions

(c) Encapsulation

Encapsulation is one of the key concepts of OOP. Encapsulation is a mechanism of binding data(variables) and code(functions) together. In Object Oriented Programming Paradigm Class is the Basic unit of a program in OOP, there by basic unit of Encapsulation. A Class binds data and code together and protects the same from outside interference and misuse by The unauthorized persons. Encapsulation is often used to hide the internal state of instances of a class from the outside interference, This is called information hiding, by using this concept we might hide the private information and also controls the access to the internal state of instances of a class. Encapsulation is going to facilitate the completeness and integrity that shows the relation of the data and methods of the classes in Integration.

Fig(1) shows the Encapsulation modeling

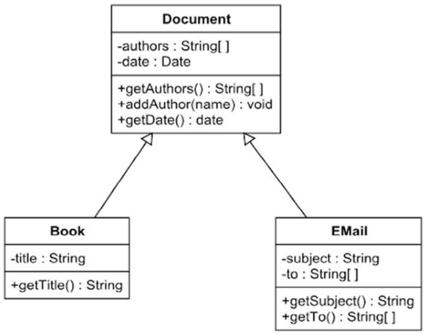
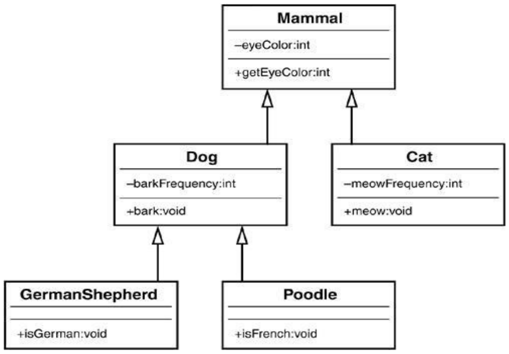


Fig. 1. Encapsulation of Data and Methods

As We can see in Fig(1)It is possible to bind all the variables and functions in classes.

(d) Inheritance

Inheritance is a important concept of OOP in which a class a can acquire the properties of another class In other words a class can acquire the state(attributes) and behaviour(functions) of another class. by using inheritance attributes and functions will be derived from base class(parent class) into the derived class(child class).The main use of Inheritance is reusability of code, due to the reusability feature of inheritance the derived class or child class gets the attributes and functions of its base class, there by Execution time will be reduced and also errors will be minimized. Reusing the attributes and functions Make the task of programming easy and the reusability of code from our existing class into our new Derived class is very feasible for the users. Inheritance has “is-a” relationship with derived classes. Inheritance also allows a class to derive its attributes and functions from more than one class/s.If changes takes place in the inherited classes, they will be operated in the other inheriting classes.



Fig(2) shows the representation of Inheritance Model

Fig. 2. Representation of Inheritance Model

As we can see in Fig. 2. child classes(Dog and Cat) can be derived from a base class named Mammal

(e) Polymorphism

Polymorphism in one another important concept of OOP.Polymorphism facilitates the users to design A common interface for a group of related actions. In other words Polymorphism allows to define one Name(object) and many forms(functions).Polymorphism makes a function to get implemented In different ways in base classes and derived classes and get different forms The OOP Programs which Implements Polymorphism will be executed fast, because of the code minimization. For example in OOP Languages, to implement stack data structure one common interface(object) is designed and that can be used for many forms(many data types),but that is not possible for the non OOP languages

(f) Abstraction

Abstraction is on top of all other features of OOP. Abstraction allows the users manage the complexity of objects. Abstraction focuses on the Essentials of objects, and hides the implementation details of objects from the outside world. More over it allows the users to Use the interfaces of those complex parts of the objects. For example We assume “Car” As an object, Car hides all the complex functionality from the user and exposes only Interfaces like steering, breaking, lights to the user(driver).In other words car object hides the Functionality of the complex parts of it from the user and giving access to interfaces of those Complex parts to the user. The difference Between Abstraction and Encapsulation is Abstraction will focuses only on the relevant details of the object and hides the unnecessary complexity of the from the user, where as Encapsulation binds and hides the data and functions into a single unit and lays a protective wrapper around it and protects the same from unauthorized access.

2.2 OBJECT ORIENTED PROGRAMMING LANGUAGES

Object Oriented Programming Languages had rich history, way back to 1960’s.The initial OOP Languages are Simula, Simula 67 and Small Talk.Small Talk was developed on the footprints of Simula 67.but Small Talk popularly know as a fully OOP Language, that means every thing is an object in Small Talk.Later on in 1980’s C++ was grown into a OOP Language. C++ is having basic programming constructs from C Language. That is why the initial name of C++ is “C with Classes.The Golden Era of Object Oriented Programming languages has been started in the 1990’s.By the invention of Java Language in 1995,the facet of OOP languages has been changed in the Software Industry. Its because the simple and powerful nature of the programming constructs of Java Language. Then Ruby and C# has got prominent role on the stage of Object Oriented Programming (OOP).Finally Python has become a Prominent Multi Paradigm Programming Language. The popularity of Python is because of its simplicity and implementation of appropriate OOP features. Python has been developed to meet the software requirements of the real world. All of these Object Oriented Programming Languages implements some or all the features of Object Oriented Programming. For example Small talk Implements almost all of the OOP features ,because every thing is an object in Small Talk. Where as Java

does not support Multiple Inheritance, one of the OOP features

The Table.(1) show the comparison of Features of OOP Languages such as Small talk,Eiffel,C++ and Java

Language	Smalltalk	Eiffel	C++	Java
Uniformity	High	Medium	Low	Medium
Documentation value	Medium	High	Medium	High
Reliability	Medium	Medium	Low	High
Protected operations	No	No	Yes	Yes
Multiple inheritance	No	Yes	Yes	No
Efficiency	Low	Medium	High	Low
Garbage collection	Yes	Yes	No	Yes
Language complexity	Low	Medium	High	Medium

Table.(1). Comparison of OOP Languages Features

2.3 OBJECT ORIENTED PROGRAMMING VS PROCEDURE ORIENTED PROGRAMMING

The Programming languages such as Cobol, Fortran and C are going to implement the features of Procedure Oriented Programming(POP) Paradigm.POP languages contains variables and procedures. These procedures are organized into functions for modularity. These functions will share the global data and communicate with each other by transferring data in between them. The main setback of POP languages is the complexity. As the size of the project grows, the complexity of that particular project will grow enormously

The Object Oriented Programming Languages like C++,Java,C# and Python came into existence to meet the requirements of the real world. The main strengths of OOP is reusability, security and Readability. Reusability of OOP reduces the redundancy of code and also reduces the production time.The data and functions of a OOP program is enclosed in a programming unit called class, there by class protects the data and functions from the unauthorized access, this enforces the security of code. The OOP programs Are easily readable and understandable, as the OOP programs are organized in the classes. In OOP.The message passing is very important concept in which messages can be communicated in between the objects.

3.MERITS OF OBJECT ORIENTED PROGRAMMING

Object Oriented Programming(OOP) is playing a major role in Software Development as one of the Programming Paradigms, It is because of its unique features such as reusability, extensibility and flexibility. Reusability of code is very useful for any Software Developer in two ways.First the time of developing the code is reduced drastically because of reusing the code again and again in the new classes. Second the reusable code is reliable as we are using the already existed code. Extensibility is a concept of Inheritance which allows a new class(derived class) to inherit the attributes and functions of a already existing class. The derived class can acquire the attributes and behaviour of its base class, there by extending the traits of base class/classes. Flexibility is one another important features of OOP. Flexibility is a concept by which a single interface can be used for the group of functions. this single interface(object) can be used for message passing for different functions for different con

Object Oriented Programming(OOP) has impacted the Software Industry and the Real World by playing a major role in the expansion of Internet and Embedded Systems. By the time of development of Java Language, the Internet has becoming popular, this popularity has been increased rapidly by the OOP Languages like Java and C#. Java also has its importance in Embedded Systems and Consumer Electronics. The initial genre of Java is Embedded Systems and Consumer Electronics. That is why Java Has been designed as a Platform Independent Programming Language. The Other important OOP Languages like Ruby and Python has also got their importance in making the Internet better.

4.ISSUES OF OBJECT ORIENTED PROGRAMMING

The following are the some of the issues related to the Object Oriented Programming

(a) Modelling Versus Reuse

During the last 10 years of change our focus is toward the object oriented community. From being mainly interested in reuse of code, the focus has shifted to analysis and design. I argue that there should be a balance between these extremes to fully realize the benefits of the potential for object orientation to integrate analysis, design and implementation

(b) Explicit Conceptual Framework

The conceptual framework underlying object oriented modelling should be used independently of the languages. This separation is necessary to avoid limiting the developer by the expressive power of the languages and to produce new requirements for language design.

(c) Abstraction Mechanisms

Although there is some procedure about the core abstraction mechanisms, there are still a number of differences between the common languages. I discuss alternative mechanisms for supporting classification and composition, single and multiple inheritance, inner versus super, generosity versus virtual classes, encapsulation, dynamic versus static typing and types versus

(d) Class-based Versus Prototype-Based Languages

Prototype-based languages are one of the most interesting developments within object-orientation. Prototype-based languages makes possible to program without classes, in general they do not support programming using a class-like style. I argue that the class-like style should be supported.

(e) Concurrency

Simula67 uses a notation of quasi-parallel processes in the form of active objects that are used for representing concurrent processes. They have also been the basis for the design of concurrency in BETA. No other Object oriented language seems to have adapted the Simula67 notation of active object. There are a number of proposals for concurrency in object-oriented languages, but no single model has been widely accepted. We argue that the Simula67 approach is considered a good approach to concurrency.

(f) Features

When people are questioned with the fundamentals of OO, they reply with a list of features provided by OOP languages instead of what OO is truly about. The features mentioned mostly are inheritance, polymorphism, encapsulation, and abstraction. For OO these features are irrelevant, apart from abstraction but the term is wrongly used here. To describe OO in terms of features provided by OOP languages that support OO leads to the conclusion that for a programming language to be OO, it has to support these features. This circular reasoning is certainly not helpful for a good understanding of what OO is truly about.

(g) Abstraction and Generalization

There is a lot of confusion over abstraction and generalization, or rather they are interchanged. But Abstraction and generalization are not the same. With abstraction some detail is left out that is considered not important in a description on a higher level of abstraction. With generalization that detail is not left out, but described in a general way on the same level of abstraction.

5. CONCLUSION

This Research on Object Oriented Programming has been the fields of the Software Development. In spite of some of the issues of Object Oriented Programming, The OOP has many advantages and merits Which are efficiently implemented in various fields of Software Industry ranging from Internet, Robotics, Gaming to Scientific Applications. By the us

of the OOP features such as Classes, Objects, Encapsulation, Polymorphism, Inheritance and Abstraction ,the Projects and Products of real time can be developed in an efficient and better way.

REFERENCES

[1] NAYER A ,F F WU and K IMHOT Object Oriented Programming for Flexible software Example of a Load flow Power Systems IEEE Transactions on 1990 5(3) :Page 689-696

[2] Katura : Object Oriented Programming and Software Engineering 1996

[3] Procedure Oriented Programming(POP) vs Object Oriented Programming(OOP). 2011.

[4] Zimmerman, M.L. and P. Mallasch. Using object-oriented Programming in computational electromagnetic codes. in Antennas and Propagation Society International Symposium, 1995. AP-S. Digest.1995.

[5] Omar N,N A Razik Determining the basic elements of Object Oriented Programming using Natural Language Processing in International Technology 2008

[6] McIntyre, S.C. and L.F. Higgins, Object-oriented Systems Analysis and Design: Methodology and Application. Journal of Management Information Systems, 1988. 5(1): p. 25-35.