Year : 2016 (FALL)

1(a) What is Object-Oriented analysis and design? Support your answer with suitable example.

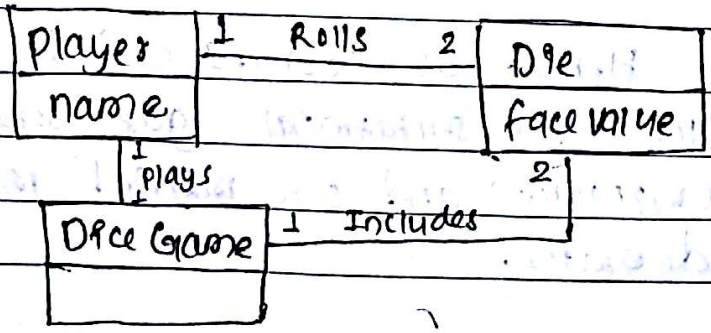
Sol) Object Oriented Analysis (OOA) is the procedure of identifying software engineering requirements and developing software specifications in terms of a software system's object model, which comprises of interacting objects. So, Object oriented analysis and design (OOAD) is a popular technical approach for analyzing and designing an application, system or business by applying object oriented programming, as well as using visual modelling throughout the development life cycles to foster better stake holder communication and product quality.

For example, consider a 'dice game' in which software simulates a player rolling two dice. If total is seven, he wins else lose.

Define use case.

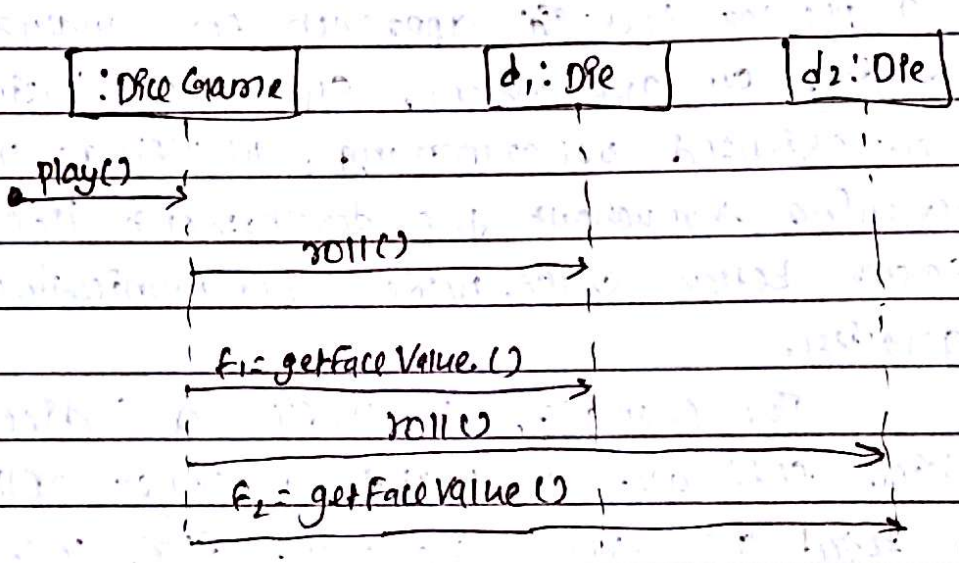
- Player requests to roll the dice. System presents results: If the dice face value totals seven, player wins; Otherwise, player loses.

Define a Domain Model

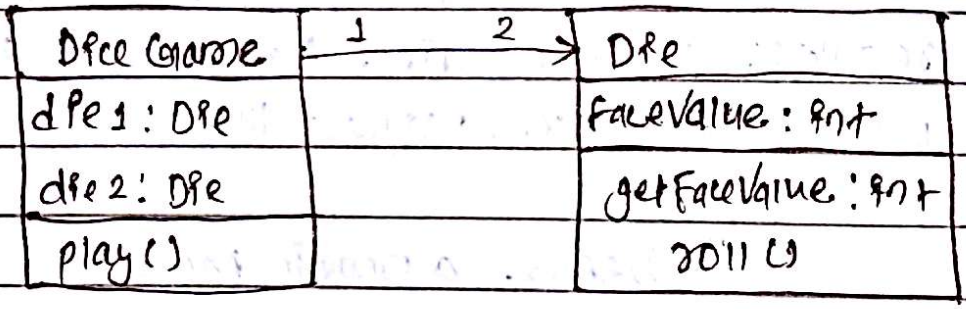


so, domain model helps in visualization of the concepts or mental models of a real-world domain. so, it is also called conceptual object model.

Assign object Responsibility and Draw interaction diagram.



Define Design class Diagram



Hence OO designs and languages can support a lower representational gap between the software components and our mental modes of a domain.

1(b) What are phases of Unified process? Describe in short.

Soln → The Unified process is a popular iterative and incremental software development process framework. It is the best known and extensively documented refinement of the Unified process. It is not just only a process but rather an extensively used framework which should be customized for specific organizations or projects. The phases of unified process are enlisted below:

(i) Inception
(ii) Elaboration
(iii) construction
(iv) Transition

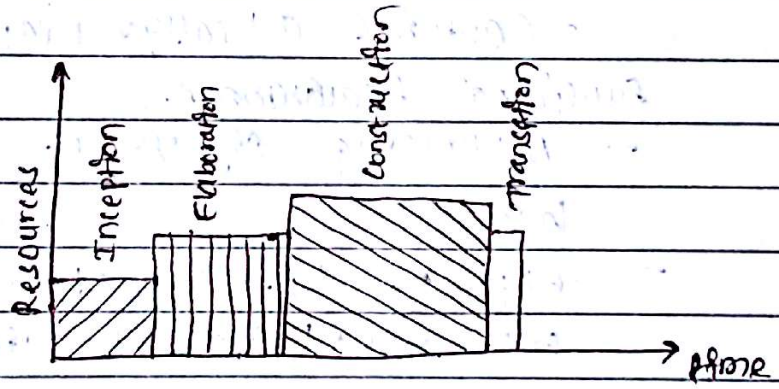


Figure: Graph showing time Vs resources in UP.

* Inception phase:

Inception is the smallest phase in project, and ideally is short phase. The following are typical goals for the inception phase:

- Establish

- prepare preliminary project schedule and cost estimate.
- Feasibility
- Buy or develop it

The life cycle objective milestone marks the end of the inception phase.

Elaboration

The goal of elaboration phase is to establish the ability to build new system given the financial constraints, and other kinds of constraints that the development project faces. The task in elaboration phase includes:

- Capturing a healthy majority of remaining functional requirement.
- Addressing significant risk on ongoing basis.
- expanding the candidate architecture into full architectural baseline.

And major milestone associated with elaboration phase are:

- most of the functional requirements for new system ~~has~~ have been captured in. in the use case modes.
- architectural baseline is complete which will serve as solid foundation for ongoing development.

Construction:

The primary goal of construction phase is to build a system capable of operating successfully in beta customer environments. During construction, the team performs tasks that involve building the system iteratively and incrementally making sure that the viability of the system is always evident in executable form.

Transition:

The primary goal of transition phase is to roll out the fully functional system to customers. During transition, the project team focuses on correcting defects and modifying the system to correct previously unidentified problems.

2 (a) "System development is model development". Do you agree? Justify.

Sol'n → The system development is process of system engineering, information system and software engineering to describe a process of planning, creating, testing and deploying an information system. Where as model development is the formulation of conceptual model as a result of system modeling that describes and represents systems.

As like in model development, first

Date
Page

Of all we specify domain for which the model is to be developed. Then, the model is developed as per the need, it is tested for it's output. Similarly a system is developed as in the same way. It goes through number of tests and improvements. As like: a model of new component is needed then it is attached, like the same way a sub-system is added into system when the new component is needed. Hence, we can state that "system development is model development."

2(b) Discuss the strength and weakness of object-oriented and prioedural programming with the help of banking transaction example.

Sol? The strength of object-oriented programming are:

- Modularity for easier troubleshooting.
- Reuse of code through inheritance
- Flexibility through polymorphism.
- Effective problem solving.

The weakness of OOP are:

- programs are of large size.
- it requires a lot of work to create a project.

- Object oriented programs are slower than other programs. programs demand more system resources, thus slowing the program down.

The strength of procedural programming are:
- Excellent for general purpose programming.
- No-need to reinvent the wheel as well tried and tested coding algorithms are available.
- Good level of control without having to know precise target CPU details.
- portable source code.

The weakness of procedural programming are:
- Lack of a cognitive structure for understanding the relationship between all the procedures and functions and data.
- programmers need to specialize in a specific procedural programming language.
- It is difficult to relate with the real world projects.
- It is difficult to maintain, if the code grows larger.
- lack of security.

For example, let us take banking transaction:

OO programming	procedural programming

main()
{
Customer c1 = New Customer(acc-no);
/* some codes */
Transaction t = New Transaction()
t.makeTransaction(c1, amount);
t.flush();
/* Some codes */
}
	main()
	{
	Struct Customer c1;
	/* some codes */
	c1.accno = acc-no;
	/* Some codes */
	makeTransn(c1, amount);
	/* some codes */
	}

3(a) Describe work flow for capturing requirement as use cases, including the participating workers and their activities.

Sol? Use case model allows developer and customer to agree on requirements, i.e. conditions and capabilities to which the system must confirm a model of system containing actors and use cases and their relationships. Actor represents user type. Each type of users may be represented by one or more actors often corresponds to worker in a business. A role of worker

defines what a worker does in a particular business process. The roles can be used to derive corresponding actors will play. Each use case represents a way the actors use the system. A use case specifies a sequence of actions, including alternatives of the sequence; that the system can perform. Example: Withdraw money (granted, denied, different amount etc). Use case instance is the execution of a usecase. It is one path through the use case. An sequence of interactions between an actor instance and the use case. Each worker in a business is an actor. During such process of enlisting actors, it should be possible to identify at least one user who can play the candidate actor- minimal overlap between roles.

Example: buyer, seller, accounting system etc

3(b) Design is four dimensional view of a system. Justify along with design concepts.

Sol? Software design is a process through which requirements are translated into a representation of software. From a project management point of view, software design can be conducted in two main steps:

① preliminary Design:

concerned with the transformation of requirements into data and software architecture.

(9) Detail Design:

Focuses on refining the architectural representation and lead to detailed data structure and algorithmic representation of software.

The three main design activities concerned in Design phase are: Data Design, Architectural Design and procedural Design. In addition, many modern applications have a distinct interface design activity.

- Data design is used to transform the information domain into data structures.
- Architectural Design is used to develop a modular program structure and represent the control relationships between modules.
- procedural design is used to transform structural components into a procedural description of the software.
- Interface Design establishes the layout and interaction mechanisms for human-machine interaction.

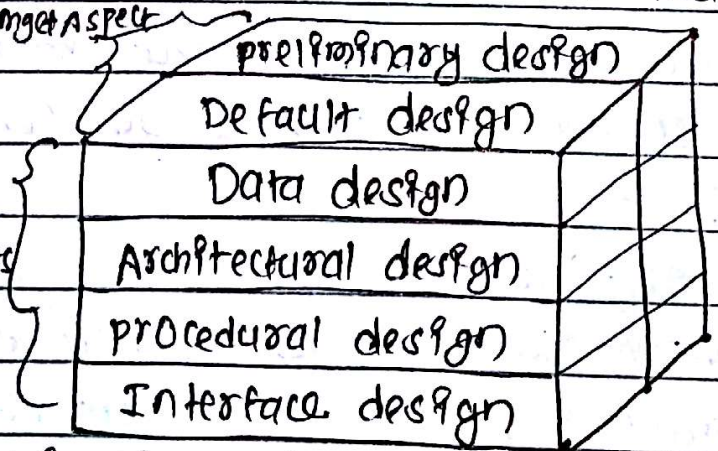


Figure: Relationship betn technical & Mgt. Aspect of design.

4@ Define design pattern. How is design pattern important? Is software development possible without applying design pattern?

Sol? → A design pattern is a general repeatable solution to a commonly occuring problem in software design. A design pattern is not finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

The design pattern are important because:

- They can speed up development process by providing tested, proven development paradigms.
- Reusing it helps to prevent subtle issues that can cause major problems and improve code readability for coders.
- They provide general solutions, documented in a format that does not require specification to a particular problem.
- Allows developers to communicate using well-known, well understood names for software interactions.

Yes, software development is possible without applying design pattern. As it is just a general solution to some common problem. But developers may choose to resolve using other methods. But using them highly helps in speeding up development.

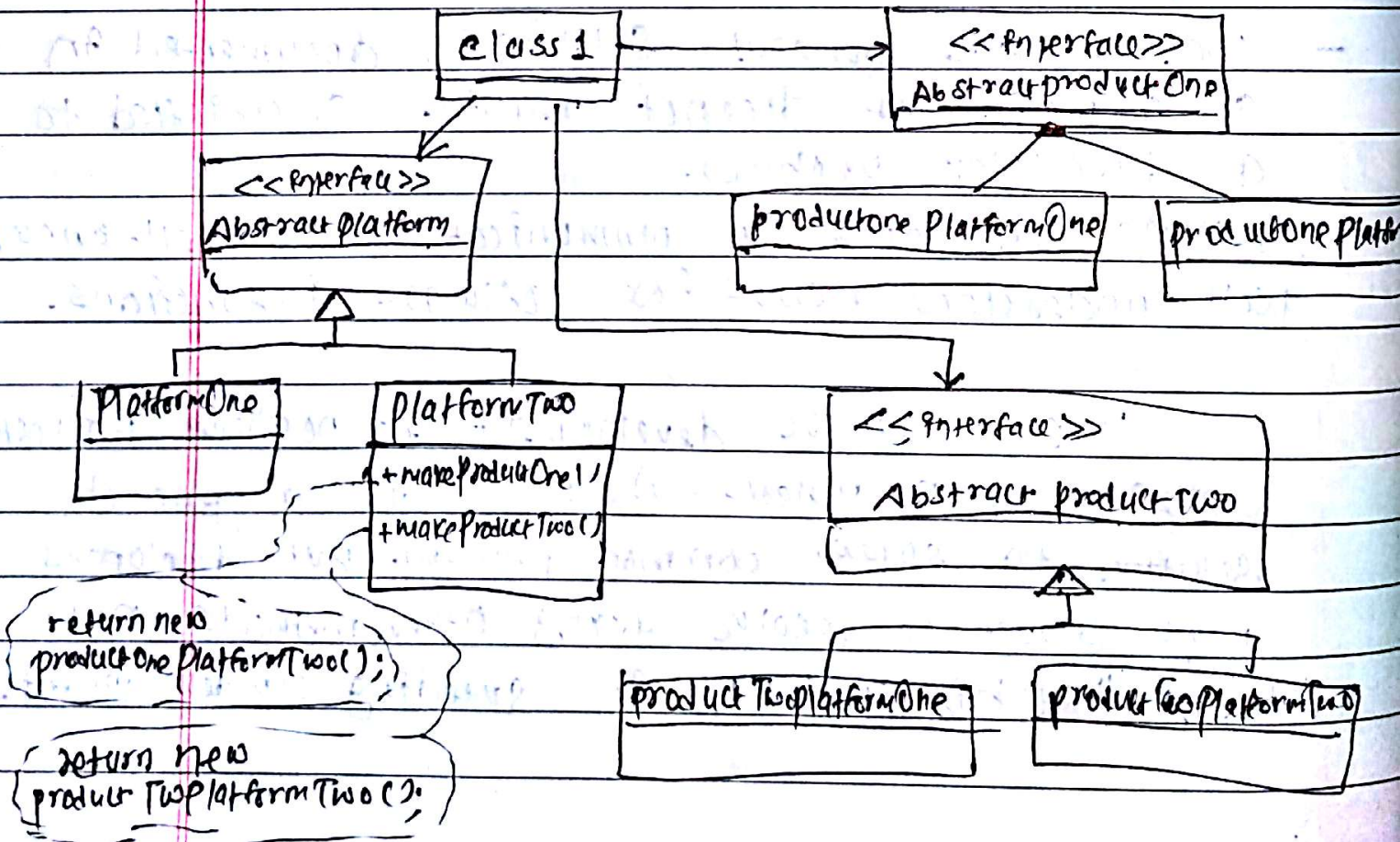
4(b) Describe suitable design pattern for following problem. "What is the best way to represent related objects (occurence) in a class diagram?"

Sol? → As looking at the problem, the best solution or design pattern is Abstract Factory. It is so because

→ It provides an interface for creating families of related or dependent objects without specifying their concrete classes.

→ A hierarchy that encapsulates: many possible "platforms and the construction of a suite of "products".

→ The new operator considered harmful, so, the structure will be,



5@ Define design principle, design concept and design pattern. What are the disadvantages of design patterns?

Sol? → Design principles represent a set of guidelines that help us to avoid having a bad design. The design principles are associated to Robert Martin who gathered them in "Agile Software Development: principles, patterns and practices". According to Robert martin there are 3 important characteristics of a bad design that should be avoided.

Rigidity:
It is hard to change because every change affects too many parts of the system.

Fragility
When you change unexpected parts of system break.

Immobility:
It is hard to reuse another application because it cannot be disentangled from current application.

Design concepts usually describes about the abstraction in system. How to provide abstraction between lower level and higher level. And hence, the types of abstraction are

① procedural Abstraction
② Data Abstraction

Design pattern is a general repetable solution, to a Commonly occurring problems in software design. It is description or template for how to solve a problem that can be used in many different situations.

The disadvantages of Design pattern are
- They do not lead to direct code reuse.
- They are complex in nature.
- They are deceptively simple
- They are validated by experience and discussion.

5(b) What is software architecture? Why do we need it?

Sol'n → Software architecture refers to the high level structures of a software system, the discipline of creating such structures, and the documentation of these structures. These structures are needed to reason about the software system. Each structure comprises software elements, relations among them, and properties of both elements and relations.

Software architecture is about making fundamental structural choices which are costly to change once implemented. Software architecture choices include specific structural options from possibilities, in the design of software. For example, the system that control the space shuttle launch vechicle had the requirement of being very fast and very reliable. Therefore an appropriate real-time computing language would need to be chosen.

We need software architecture for

- high productivity
- Better code maintainability
- higher adaptability
- Hype agnostic

6 (a) Describe software oriented architecture and design principles it helps to adhere.

— A software-oriented architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. The basic principles of service oriented architecture are independent of vendors, product and technologies. A service is a discrete unit of functionality that can be accessed remotely and acted upon

and updated independently, such as retrieving a credit card statement online. A service has four properties according to one of many definitions of SOA.

(i) It logically represents a business activity with specified outcome.

(ii) It is self-contained.

(iii) It is a black box for it's consumers.

(iv) It may consist of other underlying services.

The service oriented design principles may be broadly categorized as follows:

- Standardized service contract
- Service loose coupling
- service abstraction
- service reusability
- Service autonomy
- Service statelessness
- Service discoverability

6(b) Discuss in short about MVC Architecture with suitable diagram.

→ Model-View-Controller (MVC) architecture is software architectural pattern for implementing good software on computers. It divides a given application into three interconnected parts. This is done to separate internal

representation of information from the way information is presented to and accepted from the user. the MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.

Traditionally used for desktop GUIs, this architecture has become popular for designers in web applications and even mobile, desktop and other clients. popular programming language like Java, C#, Ruby; PHP and others have popular MVC framework that are currently being used in web application development straight out of the box.

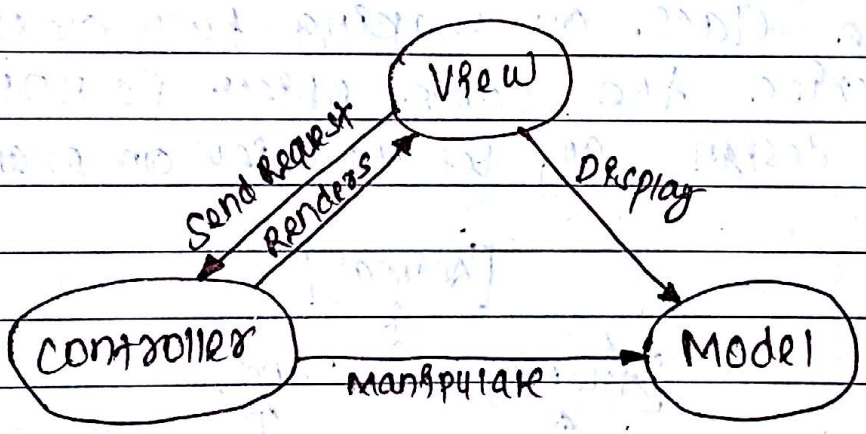


Fig: MVC Architecture

The model stores data that is retrieved according to commands from the controller and displayed in the view. A view generates new output to the user based on changes in the model. A

controller can send commands to the model's state. It can send commands to it's associated view to change the view's presentation of the model.

7 (a) Player role design pattern:

We know during development, we may need to assign different roles in different context of an application. An object may not need to play one or both roles simultaneously, so it is desirable to improve encapsulation keeping information related to only one role in one class, or making two classes to describe the same object is not good OO design so, let us view an example:

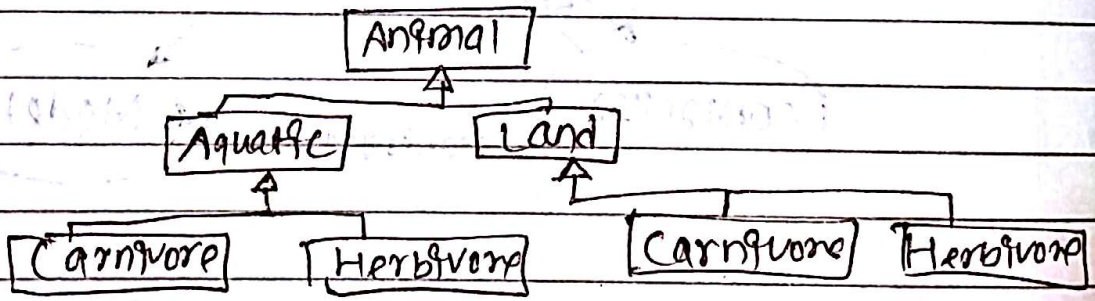


Figure: Badly designed classes.

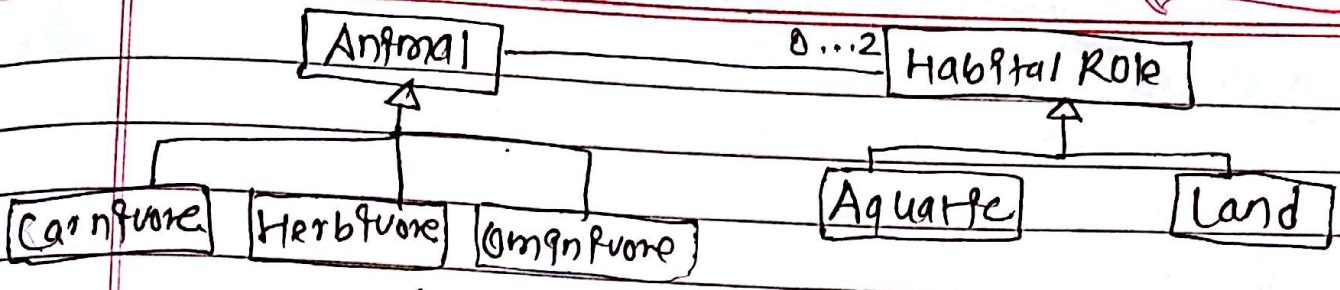
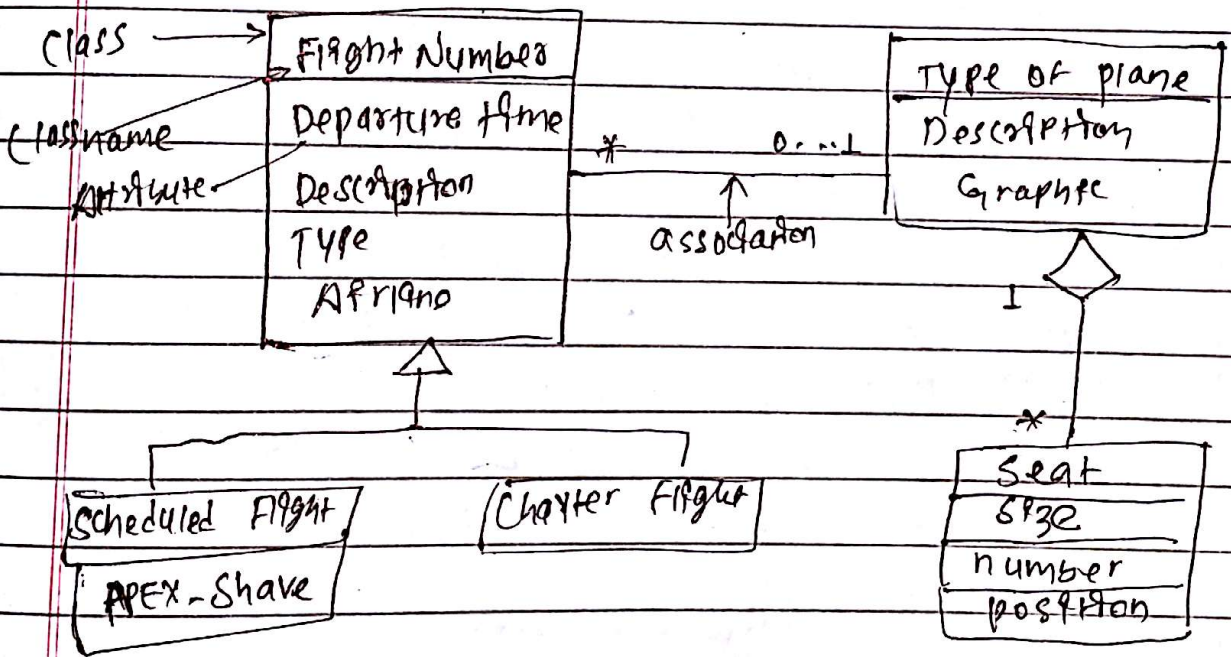


Fig: Finely designed classes.

7© Class Diagram

Class Diagram as a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.



A class represents a relevant concept from the domain, a set of persons, objects, or ideas that are depicted on the IT system.