What is a class diagram?

ASSIGN BUSTER

A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations. Basic Class Diagram Symbols and NotationsClassesClasses represent an abstraction of entities with common characteristics. Associations represent the relationships between classes. Illustrate classes with rectangles divided into compartments.

Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition (left-aligned, not bolded, and lowercase), and write operations into the third. Active ClassesActive classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border. VisibilityUse visibility markers to signify who can access the information contained within a class.

Private visibility, denoted with a - sign, hides information from anything outside the class partition. Public visibility, denoted with a + sign, allows all other classes to view the marked information. Protected visibility, denoted with a # sign, allows child classes to access information they inherited from a parent class. AssociationsAssociations represent static relationships between classes. Place association names above, on, or below the association line.

Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other. Multiplicity (Cardinality)Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one

class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for just one company. Composition and AggregationComposition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part.

Illustrate composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the " whole" class plays a more important role than the " part" class, but the two classes are not dependent on each other. The diamond ends in both composition and aggregation relationships point toward the " whole" class (i. e., the aggregation). GeneralizationGeneralization is another name for inheritance or an " is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another.

For example, Honda is a type of car. So the class Honda would have a generalization relationship with the class car. In real life coding examples, the difference between inheritance and aggregation can be confusing. If you have an aggregation relationship, the aggregate (the whole) can access only the PUBLIC functions of the part class. On the other hand, inheritance allows the inheriting class to access both the PUBLIC and PROTECTED functions of the superclass. https://www. smartdraw. com/uml-diagram/5.

Describe the six (6) different relationship notation that exists in UML Class Diagram? (6 Marks)

Answer: Relationships in Class DiagramsClasses are interrelated to each other in specific ways. In particular, relationships in class diagrams include different types of logical connections.

The following are such types of logical connections that are possible in UML:

• Association

• Directed Association

• Reflexive Association

• Multiplicity

• Aggregation

• Composition

• Inheritance/Generalization

• Realization

Associationis a broad term that encompasses just about any logical connection or relationship between classes. For example, passenger and airline may be linked as above: Directed Associationrefers to a directional relationship represented by a line with an arrowhead. The arrowhead depicts a container-contained directional flow. Reflexive AssociationThis occurs when a class may have multiple functions or responsibilities.

For example, a staff member working in an airport may be a pilot, aviation engineer, a ticket dispatcher, a guard, or a maintenance crew member. If the

maintenance crew member is managed by the aviation engineer there could be a managed by relationship in two instances of the same class. Multiplicityis the active logical association when the cardinality of a class in relation to another is being depicted. For example, one fleet may include multiple airplanes, while one commercial airplane may contain zero to many passengers. The notation 0..* in the diagram means " zero to many" Aggregationrefers to the formation of a particular class as a result of one class being aggregated or built as a collection.

For example, the class " library" is made up of one or more books, among other materials. In aggregation, the contained classes are not strongly dependent on the lifecycle of the container. In the same example, books will remain so even when the library is dissolved. To show aggregation in a diagram, draw a line from the parent class to the child class with a diamond shape near the parent class. CompositionThe composition relationship is very similar to the aggregation relationship. with the only difference being its key purpose of emphasizing the dependence of the contained class to the life cycle of the container class. That is, the contained class will be obliterated when the container class is destroyed.

For example, a shoulder bag's side pocket will also cease to exist once the shoulder bag is destroyed. Inheritance / Generalizationrefers to a type of relationship wherein one associated class is a child of another by virtue of assuming the same functionalities of the parent class. In other words, the child class is a specific type of the parent class.

To show inheritance in a UML diagram, a solid line from the child class to the parent class is drawn using an unfilled arrowhead. Realizationdenotes the implementation of the functionality defined in one class by another class. To show the relationship in UML, a broken line with an unfilled solid arrowhead is drawn from the class that defines the functionality to the class that implements the function. In the example, the printing preferences that are set using the printer setup interface are being implemented by the printer. https://creately. com/blog/diagrams/class-diagram-relationships/

6. Provide the list of six (6) " Multiplicity" constraint?

ANSWER: MultiplicityMultiplicity is a definition of cardinality - i. e. number of elements - of some collection of elements by providing an inclusive interval of non-negative integers to specify the allowable number of instances of described element. Multiplicity interval has some lower bound and (possibly infinite) upper bound: multiplicity-range ::= [lower-bound '..'] upper-bound lower-bound ::= natural-value-specification upper-bound ::= natural-value-specification | '*'Lower and upper bounds could be natural constants or constant expressions evaluated to natural (non negative) number. Upper bound could be also specified as asterisk '*' which denotes unlimited number of elements. Upper bound should be greater than or equal to the lower bound