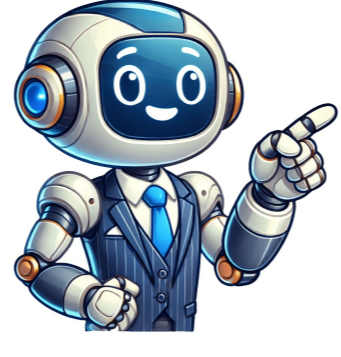


I'm human





This project is comprised of class diagrams and Java implementations for object-oriented programming exercises completed in the Object-Oriented Programming course at Universidad San Ignacio de Loyola. It focuses on creating a management system emphasizing core OOP concepts, such as relationships between classes, inheritance, and composition. Create events and assign participants. Only the organizer who created an event can modify or delete it. Participants must receive an email notification if an event they are registered for is canceled. Exercise 7: Library Management System Classes: Library: Has a name, address, and a collection of books. Book: Has title, author, publication date, and is associated with a library. User: Has a name, unique email, and a list of borrowed books. Loan: Relates a book to a user and a return date. Restrictions: A Library can have multiple books, but each Book belongs only to one library. Users can borrow multiple books, but a Book can only be associated with one Loan at a time. Only the user who borrowed the book can return it. A Book can only be deleted from the system if it is not on loan. Class diagrams are blueprints used to create objects. Unified Modelling Language (UML) is a software language that visualizes the design of a system or object. Class diagrams show classes, attributes, operations, and relationships between them, helping developers code applications. They're used for describing, visualizing, and documenting systems, and can be mapped directly with object-oriented languages. There are three main parts of a class diagram: Class Name, Class Attributes, and Class Operations. The rectangle representing the class is divided into these compartments. The class name should be written in bold at the top. When creating a class diagram, it's essential to follow specific guidelines. Start by capitalizing the first letter of each sentence. Abstract classes should be denoted in italics, highlighting their unique properties. Attributes, which are essentially the characteristics or features of an object, should be listed in the middle compartment and clearly defined. They must be meaningful and take into account the visibility factor, which determines how accessible they are. In addition to attributes, operations are also crucial components of a class diagram. These are the actions that a class can perform, corresponding to methods within the class. It's not necessary to include operations that are similar to attributes, as their relationship is implicit. Once the attributes and operations have been established, it's time to build relationships between classes. There are three primary types: generalizations, associations, and dependencies. Generalizations involve inheritance, where a subclass inherits properties from its superclass. This allows for multiple superclasses but cannot be used to model interface implementation. Associations demonstrate static relationships between entities. For instance, the association between a student and school is "studies." The multiplicity factor in associations represents how many times an attribute is multiplied. If 100 people work at an organization, then that attribute has been multiplied 100 times. Aggregation, another type of relationship, involves a whole-part connection between two classes. Dependencies show that one class relies on another, where changes to one class will affect the other. For example, an employee's dependency on their organization is clear. Class diagrams can be applied in various contexts, such as designing ATMs or hotel management systems. For designing a hospital management system, a class diagram template is available that can be customized according to specific needs. This domain model illustrates various classes such as patients, staff, and treatments, along with their relationships. A banking system class diagram displays entities like banks, ATMs, and customers as separate classes, listing attributes in the second compartment for each and linking them to show relationships. For instance, bank attributes include account number and balance. The student registration system class diagram features multiple classes, including students, accounts, course registration managers, and courses, with a linear design making it relatively simple. Registration, courses, and accounts are subclasses of the registration manager, connected using solid arrows. If the registration system operates differently, new classes can be added and the template modified easily. An airline reservation system class diagram showcases classes, structures, attributes, operations, and relationships, with main classes including reservations, passengers, ticket bookings, and employees. EdrawMax is a suitable tool for creating class diagrams, offering a library of professional examples that can be modified as needed, along with a user-friendly interface accessible to both professionals and beginners. With EdrawMax, class diagrams can be created quickly and easily, supporting documentation and visualizing system designs. The software provides a range of features, including 210+ types of diagrams, 10K+ free templates, 26K+ symbols, 10+ AI diagram generators, and 10+ export formats, making it an ideal alternative to Visio for designing class diagrams and other system models.

Class diagram problems and solutions pdf. Class diagram exercises and solutions. Problem solution diagram example. How to class diagram. Explain class diagram with example. Class diagram problems. Free body diagram problems and solutions class 11.