Inheritance and Polymorphism

# Before Class

1. Familiarise yourself with the concept of inheritance.
   1. <https://www.w3schools.com/java/java_inheritance.asp>
   2. <https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html>
   3. <https://www.tutorialspoint.com/java/java_inheritance.htm>
2. Answer the questions:
   1. What is class inheritance. - Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.
   2. What are the differences between superclass and subclass. - The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).
   3. What is the main purpose of class inheritance. - Inheritance enables you to create new classes that reuse, extend, and modify the behavior defined in other classes
   4. How inheritance is implemented in a programming language.
3. How the constructor of the parent class is called.

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. Familiarise yourself with the concept of polymorphism. - Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.
   1. <https://www.w3schools.com/java/java_polymorphism.asp>
   2. <https://www.tutorialspoint.com/java/java_polymorphism.htm>
2. What is method overriding - In Java, method overriding occurs when a subclass (child class) has the same method as the parent class. In other words, method overriding occurs when a subclass provides a particular implementation of a method declared by one of its parent classes.

# During Class

1. Create a project where you define a Book class that describes books. Add book attributes: title and author. Create a constructor where you assign book attribute values. Add accessor and mutator methods for all book attributes. Add also a display() method to display information about the book. Then write a program that creates and displays two books.
2. Add an Ebook class definition to your project that derives from the Book class (inherits the Book class attributes and methods). Add an attribute: file name. Create a constructor where you assign the ebook attribute values (title, author and file name). Add accessor and mutator methods for all ebook attributes. Also add a display() method to display information about the ebook. Then write a program that creates and displays one book and two e-books.
3. Add an Audiobook definition to the project that derives from the Book class (inherits the Book class's attributes and methods). Add the following attributes in the class: minutes and seconds, which define the duration of the audiobook. Create a constructor where you assign audiobook attribute values (title, author, and audiobook duration). Add accessor and mutator methods for all the audiobook attributes. Also add a display() method to display information about the audiobook. Then write a program that creates and displays one book and two audiobooks.
4. Add a Library class definition to your project that contains an attribute that stores any books, ebooks, or audiobooks. Apply an array. Consider the correct data type for the array. Then write a program that creates and displays one book, two e-books and two audiobooks. Save all object in the library. Display the contents of the library. Notice which display() object methods are called.

# After Class

1. Add a Publisher class definition to your project that includes the name and city attributes. Add a constructor in the class where you assign initial values for the attributes. Also add accessor and mutator methods for all attributes.
2. Modify the Book class by adding the publisher attribute. Then make the necessary changes to the remaining classes to display the library content along with publisher information.
3. In the UML class diagram, show all the classes defined in the project and the relationships between the classes.
4. Modify the Book class. Add the book's year of publication attribute, as well as accessor and mutator methods for this attribute. Then modify other project classes and display the library contents.
5. Modify the UML class diagram by supplementing the diagram with the year of the book's publication.
6. Define a Writer class in your project that describes the writer by his/her first name, last name and literary genre. Then, in the Book class, use an object of that class to describe the author of the book. Modify other project classes and display the library contents.
7. Modify the UML class diagram by supplementing the diagram with the Writer class and relations between other classes of the project.