



Republic of the Philippines
SURIGAO DEL NORTE STATE UNIVERSITY
Narciso Street, Surigao City 8400, Philippines



"For Nation's Greater

In Partial Fulfillment of the Requirements for the
CS 223 - Object-Oriented Programming

Theater Heritage - Play and Musical Class Structure

Presented to:
Dr. Unife O. Cagas
Professor

Prepared by:
Reyes, Jessa B.
Student

BSCS-2A Computer Science

May 2024



"For Nation's Greater

Republic of the Philippines
SURIGAO DEL NORTE STATE UNIVERSITY
Narciso Street, Surigao City 8400, Philippines



Project Title:

Theater Heritage - Play and Musical Class Structure

Project Description:

Theater history is made up of an extensive variety of dramatic works that have influenced culture over time, from classic tragedies to vibrant comedies. The code that follows reflects this legacy by using an Object-Oriented Programming (OOP) model to provide a structured play and musical representation. Using fundamental OOP concepts like polymorphism, inheritance, abstraction, and encapsulation, it creates a hierarchy of classes to capture the spirit of plays.

Objectives:

The objective of this code is to model Play and Musical equipment using an Object-Oriented Programming approach. This approach aims to:

- ❶ To use theater as a contextual framework for teaching OOP concepts.
- ❷ To establish a class system that faithfully replicates the elements and framework of musicals and plays.
- ❸ To show how OOP ideas like composition, polymorphism, inheritance, and encapsulation may be used in practical situations.

Importance and Contribution of the Project:

This code is useful for anyone who wants to understand how theatrical productions work together. This code provides a basic framework for classifying and representing different kinds of plays and musicals using Object-Oriented Programming (OOP) ideas. It distinguishes clearly between general characteristics (title, playwright, genre) and particulars (composer for musicals, protagonist for comedies). This division creates a clean, well-organized framework for modeling various theatrical genres and makes the code easier to read and maintain.

Hardware & Software Used:

Hardware:

- Computer or Cellphone

Software:

- GDB



Five Principles of Object Oriented Programming

Inheritance:

```
class Musical(Play):
```

The Musical class inherits from the Play class. This means Musical inherits all the attributes and methods defined in Play.

```
class Comedy(Play):
```

Comedy is declared as a subclass of Play. This indicates that Comedy inherits all the attributes (such as title, playwright, and genre) and methods (like display_info) defined in the Play class.

Polymorphism:

```
def display_info(self):
```

The display_info method was originally defined in the Play superclass, but it is clearly polymorphic in the way it is overridden in the Musical and Comedy subclasses.

Encapsulation:

```
def __init__(self, title, playwright, genre):
```

The __init__ method initializes instance variables (title, playwright, genre) within the Play class. These variables are encapsulated within the class and can only be accessed or modified through methods defined within the class.

Abstract:

```
def display_info(self):
```

The display_info() method in the Play class demonstrates abstraction. It hides the internal details of how play information is retrieved and formatted, providing a straightforward interface for users to access and display information about a play.



Code Documentation:

Abstraction: Using simple and clear class interfaces to hide complex details from the user

class Play:

Encapsulation: Using the constructor to initialize instance variables

```
def __init__(self, title, playwright, genre):  
    self.title = title # Encapsulation: These are instance variables  
    self.playwright = playwright # Encapsulation: These are instance variables  
    self.genre = genre # Encapsulation: These are instance variables
```

Abstraction: Providing a simple method to display play information

```
def display_info(self):  
    print(f"Title: {self.title}")  
    print(f"Playwright: {self.playwright}")  
    print(f"Genre: {self.genre}")
```

Inheritance: Musical is a subclass of Play, inheriting its properties and methods

class Musical(Play):

```
def __init__(self, title, playwright, genre, composer):  
    super().__init__(title, playwright, genre)  
    self.composer = composer # Encapsulation: This is an instance variable specific to Musical
```

Polymorphism: Overriding the display_info method to include composer information

```
def display_info(self):  
    super().display_info()  
    print(f"Composer: {self.composer}")
```

Inheritance: Comedy is a subclass of Play, inheriting its properties and methods

class Comedy(Play):

```
def __init__(self, title, playwright, genre, main_character):  
    super().__init__(title, playwright, genre)  
    self.main_character = main_character # Encapsulation: This is an instance variable  
specific to Comedy
```

Polymorphism: Overriding the display_info method to include main character information

```
def display_info(self):
```



```
super().display_info()  
print(f"Main Character: {self.main_character}")
```

Classes and Objects: Creating objects (instances of the classes)

```
play1 = Play("Hamlet", "William Shakespeare", "Tragedy") # Object instance of Play  
musical1 = Musical("The Lion King", "Roger Allers", "Musical", "Elton John") # Object instance  
of Musical  
comedy1 = Comedy("The Importance of Being Earnest", "Oscar Wilde", "Comedy", "Jack  
Worthing") # Object instance of Comedy
```

Access methods

```
play1.display_info()  
print()  
musical1.display_info()  
print()  
comedy1.display_info()
```

User Guide:

Step 1:

- Start the program by pressing the  Run button.



Step 2:

- You'll see the Program Finished.

```
...Program finished with exit code 0  
Press ENTER to exit console.
```



Output:

```
Title: Hamlet
Playwright: William Shakespeare
Genre: Tragedy

Title: The Lion King
Playwright: Roger Allers
Genre: Musical
Composer: Elton John

Title: The Importance of Being Earnest
Playwright: Oscar Wilde
Genre: Comedy
Main Character: Jack Worthing
```

Description:

This code explores the field of theater shows while entertainingly showcasing the fundamentals of Object-Oriented Programming (OOP). Through inheritance, it creates a strong foundation by defining common features such as title, playwright, and genre in a base 'Play' class. The ability to reuse components makes it easier to create specialized subclasses like Musical and Comedy. The 'Play' class makes use of abstraction to emphasize important features while keeping implementation details hidden. One example of abstraction is the 'display_info' method in 'Play', which offers a straightforward manner of displaying information. This is overridden by subclasses, which show polymorphism by providing customized descriptions. Similar to how a single key can open many locks, the 'display_info' method adjusts to offer particular details for each subclass when it is used on different instances (Play, Musical, Comedy).

Conclusion:

In summary, this code uses a realistic example focused around theater shows to successfully illustrate important Object-Oriented Programming (OOP) concepts. The fundamental 'Play' class defines common properties, and specialized subclasses like 'Musical' and 'Comedy' augment this functionality. This offers a flexible and reusable structure through inheritance. By keeping complicated details hidden and emphasizing key characteristics, abstraction is accomplished, and polymorphism is illustrated by the 'display_info' method, which changes its behavior to show different information depending on the subclass. This helps people who are unfamiliar with OOP learn about the practical application of fundamental principles while also making understanding of different play types.

References:

Online GDB Link: <https://onlinegdb.com/iTSsq0VF>

Encapsulation: <https://www.tutorialspoint.com/>

Inheritance: <https://www.w3schools.com/>

Polymorphism: <https://www.w3schools.com/>

Abstract: <https://www.tutorialspoint.com>



Republic of the Philippines
SURIGAO DEL NORTE STATE UNIVERSITY
Narciso Street, Surigao City 8400, Philippines



"For Nation's Greater
