Abstraction: YouTube Videos

• Scenario

You have been hired by a company that does product awareness monitoring by tracking the placement of their products in YouTube videos. They want you to write a program that can help them work with the tens of thousands of videos they have identified as well as the comments on them.

• Program Specification

Write a program to keep track of YouTube videos and comments left on them. As mentioned this could be part of a larger project to analyze them, but for this assignment, you will only need to worry about storing the information about a video and the comments.

Your program should have a class for a Video that has the responsibility to track the title, author, and length (in seconds) of the video. Each video also has responsibility to store a list of comments, and should have a method to return the number of comments. A comment should be defined by the Comment class which has the responsibility for tracking both the name of the person who made the comment and the text of the comment.

Once you have the classes in place, write a program that creates 3-4 videos, sets the appropriate values, and for each one add a list of 3-4 comments (with the commenter's name and text). Put each of these videos in a list.

Then, have your program iterate through the list of videos and for each one, display the title, author, length, number of comments (from the method) and then list out all of the comments for that video. Repeat this display for each video in the list.

Note: The YouTube example is just to give you a context for creating classes to store information. You will not actually be connecting to YouTube or downloading content in any way.

object: Video

Responsibility:

- To track the title, author, and length (in seconds) of the video.
- Store a list of comments, and should have a method to return the number of comments.

Behaviors:

GetNumberOfComments

State:

_title
_author
_length
_comments

object: Comment

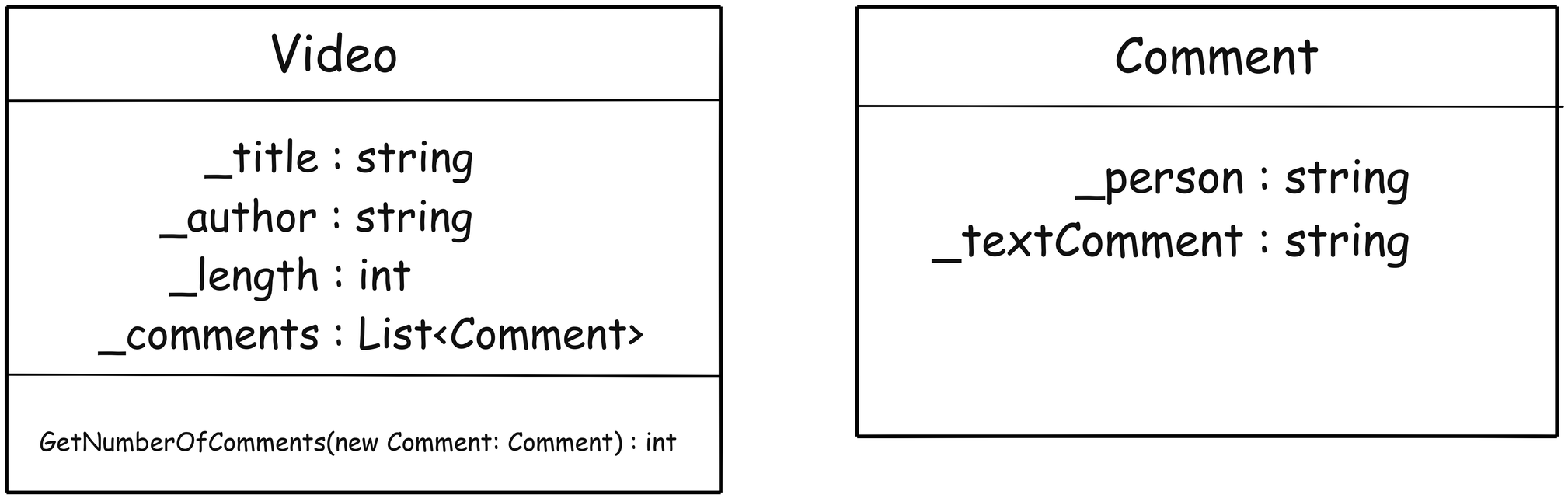
Responsibility:

- Tracks the name of the person who made the comment and the text of the comment the number of comments.

State:

_person
_textComments

Abstraction: YouTube Videos



Encapsulation: Online Ordering

• Scenario

You have been hired to help a company with their product ordering system. They sell many products online to a variety of customers and need to produce packing labels, shipping labels, and compute final prices for billing.

• Program Specification

Write a program that has classes for Product, Customer, Address, and Order. The responsibilities of these classes are as follows:

Order

Contains a list of products and a customer. Can calculate the total cost of the order. Can return a string for the packing label. Can return a string for the shipping label.

The total price is calculated as the sum of the total cost of each product plus a one-time shipping cost.

This company is based in the USA. If the customer lives in the USA, then the shipping cost is $5. If the customer does not live in the USA, then the shipping cost is $35.

A packing label should list the name and product id of each product in the order.

A shipping label should list the name and address of the customer

Product

Contains the name, product id, price, and quantity of each product.

The total cost of this product is computed by multiplying the price per unit and the quantity. (If the price per unit was $3 and they bought 5 of them, the product total cost would be $15.)

Customer

The customer contains a name and an address.

The name is a string, but the Address is a class.

The customer should have a method that can return whether they live in the USA or not. (Hint this should call a method on the address to find this.)

Address

The address contains a string for the street address, the city, state/province, and country.

The address should have a method that can return whether it is in the USA or not.

The address should have a method to return a string all of its fields together in one string (with newline characters where appropriate)

Other considerations

Make sure that all member variables are private and getters, setters, and constructors are created as needed.

Once you have created these classes, write a program that creates at least two orders with a 2-3 products each. Call the methods to get the packing label, the shipping label, and the total price of the order, and display the results of these methods.

object: Customer

Responsibility:

- Contains a name and an address.
- The name is a string, but the Address is a class.
- Should have a method that can return whether they live in the USA or not. (Hint this should call a method on the address to find this.)

Behaviors:

IsInUSA (from Address class)

State:

Customer(name, Address address)

object: Address

Responsibility:
- Contains a string for the street address, the city, state/province, and country.
- Has a method that can return whether it is in the USA or not.
- Has a method to return a string all of its fields together in one string (with newline characters where appropriate)

Behaviors:

IsInUSA

GetAddress

State:

_address

_city

_provinceOrState

_country

object: Product

Responsibility:
- Contains the name, product id, price, and quantity of each product.
- The total cost of this product is computed by multiplying the price per unit and the quantity.
(If the price per unit was $3 and they bought 5 of them, the product total cost would be $15.)

State:

_name
_productID
_price
_quantity

object: Order

Responsibility:
- Contains a list of products and a customer.
- Can calculate the total cost of the order.
- Can return a string for the packing label.
- Can return a string for the shipping label.

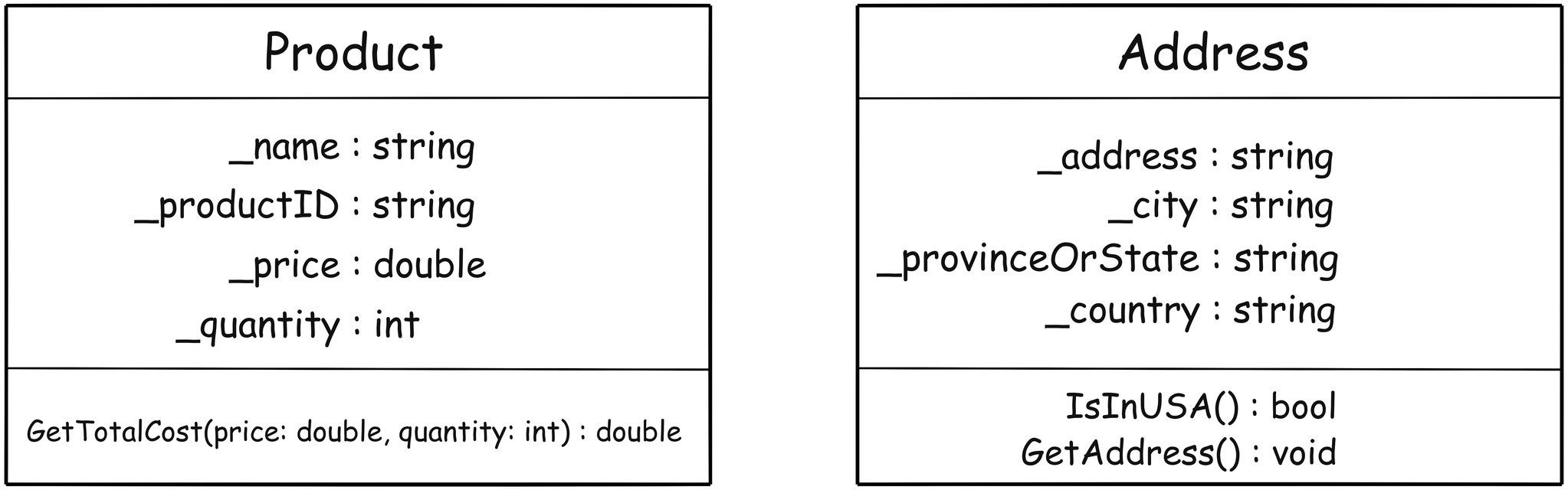
Behaviors:

GetTotalCost
PackingLabel
ShippingLabel

State:

_packingLabel
_shippingLabel
_totalOrderCost
_products
_customer

Encapsulation: Online Ordering



Inheritance: Event Planning

• Scenario

You have been hired by an event planning company. They help organize and market events throughout the world. They need you to write a program to track each of these events and produce the marketing material to distribute on social media. They typically handle a few main types of events:

Lectures, which have a speaker and have a limited capacity.

Receptions, which require people to RSVP, or register, beforehand.

Outdoor gatherings, which do not have a limit on attendees, but need to track the weather forecast.

Regardless of the type, all events need to have an Event Title, Description, Date, Time, and Address.

They would like the ability to generate three different messages:

Standard details - Lists the title, description, date, time, and address.

Full details - Lists all of the above, plus type of event and information specific to that event type. For lectures, this includes the speaker name and capacity. For receptions this includes an email for RSVP. For outdoor gatherings, this includes a statement of the weather.

Short description - Lists the type of event, title, and the date.

• Program Specification

Write a program that has a base Event class along with derived classes for each type of event. These classes should contain the necessary data and provide methods to return strings for each of the messages the company desires.

Remember that any data or methods that are common among all types of events should be in the base class.

Once you have the classes in place, write a program that creates at least one event of each type and sets all of their values. Then, for event event, call each of the methods to generate the marketing messages and output their results to the screen.

In addition, your program must:

Use inheritance to avoid duplicating shared attributes and methods.

Use an address class for the addresses.

Follow the principles of encapsulation, making sure each member variable is private.

Inheritance: Event Planning

Event

_title : string
_description : string
_date : string
_time : string
_address : string

Event()
FullDetails(title, description, date, time, address) : string
StandardDetails(title, description, date, time, address) : string
ShortDescription(type, event, title, date) : string

LectureEvent

_speaker : string
_capacity : int

*FullDetails(title, description, date, time, address, speaker, capacity: int) : string

ReceptionEvent

_email : string
_rsvp : string

*FullDetails(title, description, date, time, address, email, rsvp) : string

OutdoorEvent

_weatherStatement : string

*FullDetails(title, description, date, time, address, weatherStatement) : string

*Note: All parameters are strings, unless indicated otherwise

Polymorphism : Exercise Tracking

• Scenario

The local fitness center has hired you to write an app for their customers to track their exercise. They have facilities for the following:

Running

Stationary Bicycles

Swimming in the lap pool

For each activity, they want to track the the date and the length of the activity in minutes. Then, for each activity, they would like to also track the following:

Running: distance

Cycling: speed

Swimming: number of laps

For each activity, they do not want to store this information, but they would like to be able to get following information (by calculation if it is not stored directly):

The distance

The speed (miles per hour or kilometers per hour)

The pace (minutes per mile or minutes per kilometer)

A summary in the form of:

03 Nov 2022 Running (30 min)- Distance 3.0 miles, Speed 6.0 mph, Pace: 10.0 min per mile

03 Nov 2022 Running (30 min): Distance 4.8 km, Speed: 9.7 kph, Pace: 6.25 min per km

You may choose if your program uses miles or kilometers (you do not need to handle both). In either case the length of a lap in the lap pool is 50 meters.

• Program Specification

Write a program that has a base Activity class and then has a derived class for each of the three activities. The base class should contain any attributes that are shared among all activities. Then, each derived class can define any additional attributes.

In addition, the base class should contain virtual methods for getting the distance, speed, pace. These methods should be overridden in the derived classes.

Finally, you should provide a GetSummary method to produce a string with all the summary information.

Remember that the summary method can make use of the other methods to produce its result. This method should be available for all classes, so it should be defined in the base class (you can override it in the derived classes if needed, but it may not need to be...).

Once you have the classes in place, write a program that creates at least one activity of each type. Put each of these activities in the same list. Then iterate through this list and call the GetSummary method on each item and display the results.

In addition, your program must:

Use inheritance to avoid duplicating shared attributes and methods.

Use method overriding for the calculation methods.

Follow the principles of encapsulation, making sure each member variable is private.

Math Hints:

Distance (km) = swimming laps * 50 / 1000

Distance (miles) = swimming laps * 50 / 1000 * 0.62

Speed (mph or kph) = (distance / minutes) * 60

Pace (min per mile or min per km)= minutes / distance

Speed = 60 / pace

Pace = 60 / speed

Polymorphism : Exercise Tracking

Activity

_name : string
_date : string
_length : int

Activity()

GetSummary (new Activity activity) : string, virtual

Program.cs

_activity : List<Activity>

RunningActivity

_distance : double
_speed : double
_pace : double

GetSummary (name: string, date : string, Length : int, distance: double, speed : double, pace : double) : string, override

CyclingActivity

_distance : double
_speed : double
_pace : double

GetSummary (name: string, date : string, Length : int, distance: double, speed : double, pace : double) : string, override

SwimmingActivity

_distance : double
_speed : double
_pace : double

GetSummary (name: string, date : string, Length : int, distance: double, speed : double, pace : double) : string, override