**Assignment: Multiple Classes with Association**



Session: 2021 – 2025

**Submitted by:**

Muhammad Abdullah 2021-CS-82

**Supervised by:**

DR. Awais Hasan

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**Important Instructions**

**Here you can find the major parts of your Assignment documentation**

* Table of Contents
* Step 1: Case Study Scenario
* Step 2: Domain Model with only the Class names
* Step 3: Domain Model with Relations and Constraints
* Step 4: Domain Model with Multiplicity
* Step 5: Class Diagram

1) Attributes and

2) Functions

* ~~Step 6: Separate code of the Classes attributes to show the relations added in the attributes~~
* Step 7: Full Code

**Formatting Instructions**

1. Heading Size is 16
2. Sub heading size is 14
3. Further heading size is 13
4. Make your heading font bold
5. Text Font size is 12
6. Use Times New Roman Font Style
7. Text paragraphs should be justified. (Justify is feature of MS World)
8. Domain Model should be drawn on word or PowerPoint or some drawing tool (Hand drawn domain model will not be accepted).
9. Class Diagram should be drawn using the Visual Studio Class Designer.
10. Code Size should be 10 and 1.0 line Spacing to make it compact
11. Follow proper coding Styles to make the classes and driver program
12. Apply all these 7 steps to all your Case Studies
    1. UAMS
    2. Ocean Navigation
    3. Point and Line
    4. Point of Sales Application
    5. Telsa’s Coffee Shop
    6. GameObjects

**Step 1: Case Study Scenario**

In ocean navigation, locations are measured in degrees and minutes of latitude and longitude. Thus, if you’re lying off the mouth of Papeete Harbor in Tahiti, your location is 149 degrees 34.8 minutes west longitude, and 17 degrees 31.5 minutes south latitude. This is written as 149°34.8’ W, 17°31.5’ S. There

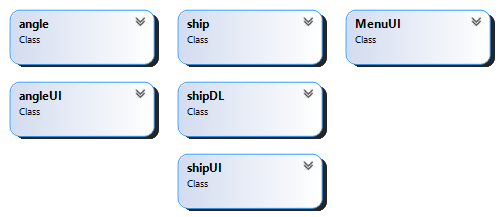
are 60 minutes in a degree. (An older system also divided a minute into 60 seconds, but the modern

approach is to use decimal minutes instead.) Longitude is measured from 0 to 180 degrees, east or west

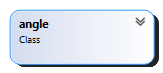
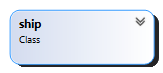
from Greenwich, England, to the international dateline in the Pacific. Latitude is measured from 0 to 90

degrees, north or south from the equator to the poles.

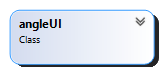
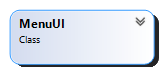
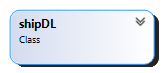
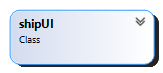
**Step 2: Domain Model with Only the Class Names**



**Step 3: Domain Model with Relations and Constraints**



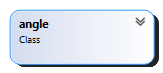
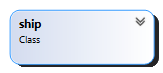
Consists



0>=Latitude<=90

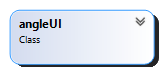
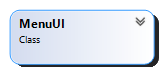
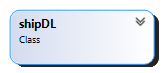
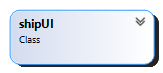
0>=Longitude<=180

Contains

**Step 4: Domain Model with Multiplicity**

∞

1



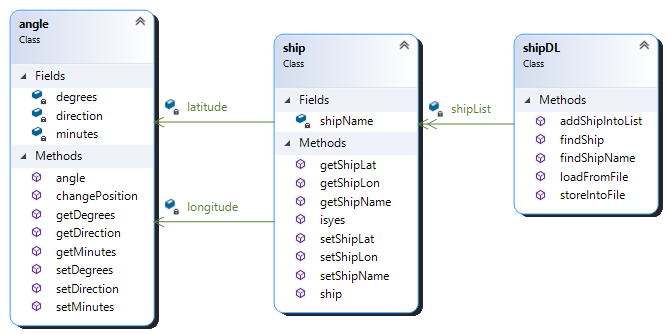
∞

1

**Step 5: Class Diagram Attributes and Functions**

0>=Latitude<=90

0>=Longitude<=180



Contains

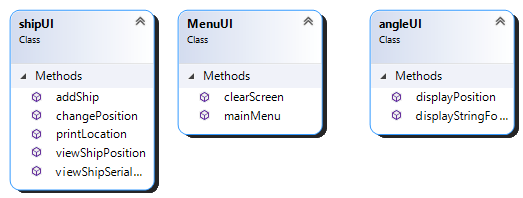
∞

1

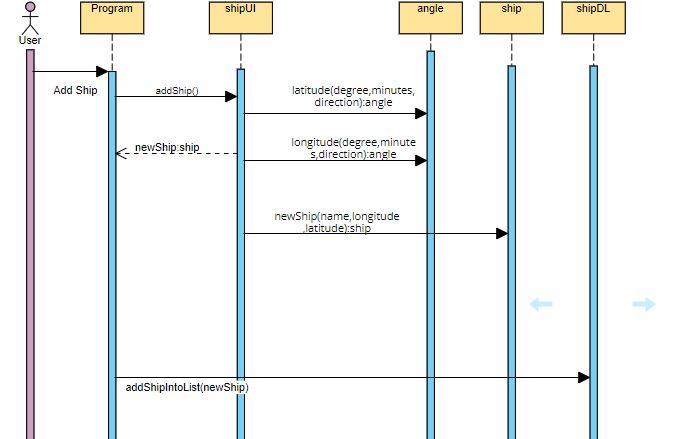
∞

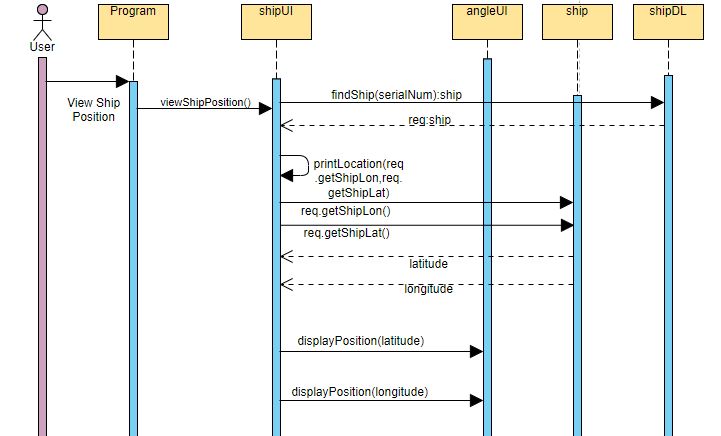
1

Consists

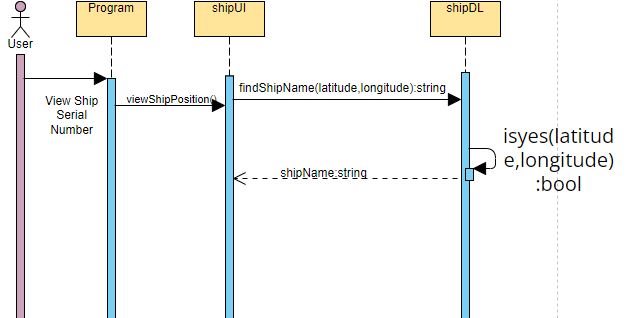


**Step 6: Sequence Diagram**

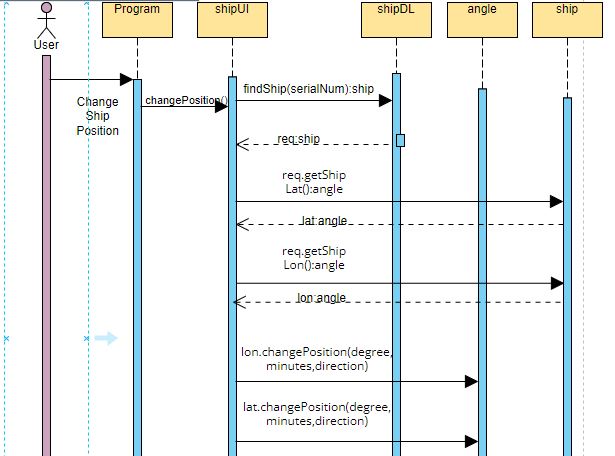
**Option 1: Add Ship:**

**Option 2: View Ship Position:**

**Option 3: View Ship Serial Number** :



**Option 4: Change Ship Position:**

****

**BL Code: angle.cs**

class angle

   {

       private int degrees;

       private float minutes;

       private char direction;

       public angle (int degrees , float minutes , char direction)

       {

           this.degrees = degrees;

           this.minutes = minutes;

           this.direction = direction;

       }

       public void changePosition (int degrees , float minutes , char direction)

       {

           this.degrees = degrees;

           this.minutes = minutes;

           this.direction = direction;

       }

       public int getDegrees ()

       {

           return degrees;

       }

       public float getMinutes ()

       {

           return minutes;

       }

       public char getDirection ()

       {

           return direction;

       }

       public void setDegrees (int degrees)

       {

           this.degrees = degrees;

       }

       public void setMinutes (float minutes)

       {

           this.minutes = minutes;

       }

       public void setDirection (char direction)

       {

           this.direction = direction;

       }

   }

**BL Code: ship.cs**

class ship

   {

       private string shipName;

       private angle longitude;

       private angle latitude;

       public ship (string shipName , angle longitude , angle latitude)

       {

           this.shipName = shipName;

           this.longitude = longitude;

           this.latitude = latitude;

       }

       public bool isyes (string latitude , string longitude)

       {

           string lon = angleUI.displayStringFormat(this.longitude);

           string lat = angleUI.displayStringFormat(this.latitude);

           if (latitude == lat && longitude == lon)

           {

               return true;

           }

           return false;

       }

       public string getShipName ()

       {

           return shipName;

       }

       public angle getShipLat ()

       {

           return latitude;

       }

       public angle getShipLon ()

       {

           return longitude;

       }

       public void setShipName (string shipName)

       {

           this.shipName = shipName;

       }

       public void setShipLat (angle latitude)

       {

           this.latitude = latitude;

       }

       public void setShipLon (angle longitude)

       {

           this.longitude = longitude;

       }

   }

**DL Code: shipDL.cs**

class shipDL

   {

       private static List<ship> shipList = new List<ship>();

       public static void addShipIntoList (ship newShip)

       {

           shipList.Add(newShip);

       }

       public static ship findShip (string serialNum)

       {

           foreach (ship item in shipList)

           {

               if (serialNum == item.getShipName())

               {

                   return item;

               }

           }

           return null;

       }

       public static string findShipName (string latitude , string longitude)

       {

           foreach (ship item in shipList)

           {

               if (item.isyes(latitude , longitude))

               {

                   return item.getShipName();

               }

           }

           return null;

       }

       public static void storeIntoFile (string path , ship newShip)

       {

           StreamWriter f = new StreamWriter(path , true);

           f.WriteLine($"{newShip.getShipName()},{newShip.getShipLon().getDegrees()},{newShip.getShipLon().getMinutes()},{newShip.getShipLon().getDirection()},{newShip.getShipLat().getDegrees()},{newShip.getShipLat().getMinutes()},{newShip.getShipLat().getDirection()}");

           f.Flush();

           f.Close();

       }

       public static void loadFromFile (string path)

       {

           StreamReader f = new StreamReader(path);

           string record;

           while ((record = f.ReadLine()) != null)

           {

               string[] splittedrecord = record.Split(',');

               string shipName = splittedrecord[0];

               int degrees = int.Parse(splittedrecord[1]);

               float minutes = float.Parse(splittedrecord[2]);

               char direction = char.Parse(splittedrecord[3]);

               angle longitude = new angle(degrees , minutes , direction);

               degrees = int.Parse(splittedrecord[4]);

               minutes = float.Parse(splittedrecord[5]);

               direction = char.Parse(splittedrecord[6]);

               angle latitude = new angle(degrees , minutes , direction);

               ship newShip = new ship(shipName , longitude , latitude);

               shipList.Add(newShip);

           }

           f.Close();

       }

   }

**UI Code: shipUI.cs**

class shipUI

    {

      public static ship addShip ()

        {

            bool flag = true;

            Console.WriteLine("Enter Ship Serial Number");

            string name = Console.ReadLine();

            Console.WriteLine("Enter Ship Latitude:");

            int degree = 0;

            while (flag)

            {

                Console.WriteLine("Enter Latitude's Degree:");

                degree = int.Parse(Console.ReadLine());

                if (degree < 0 || degree > 90)

                {

                    Console.WriteLine("Enter Again");

                }

                else

                {

                    flag = false;

                }

            }

            flag = true;

            Console.WriteLine("Enter Latitude's Minutes:");

            float minutes = float.Parse(Console.ReadLine());

            Console.WriteLine("Enter Latitude's Direction:");

            char direction = char.Parse(Console.ReadLine());

            angle latitude = new angle(degree , minutes , direction);

            Console.WriteLine("Enter Ship Longitude:");

            while (flag)

            {

                Console.WriteLine("Enter Longitude's Degree:");

                degree = int.Parse(Console.ReadLine());

                if (degree < 0 || degree > 180)

                {

                    Console.WriteLine("Enter Again");

                }

                else

                {

                    flag = false;

                }

            }

            Console.WriteLine("Enter Longitude's Minutes:");

            minutes = float.Parse(Console.ReadLine());

            Console.WriteLine("Enter Longitude's Direction:");

            direction = char.Parse(Console.ReadLine());

            angle longitude = new angle(degree , minutes , direction);

            ship newShip = new ship(name , longitude , latitude);

            return newShip;

        }

        public static void printLocation (angle longitude , angle latitude)

        {

            Console.WriteLine($"Ship is at ");

            angleUI.displayPosition(longitude);

            angleUI.displayPosition(latitude);

        }

        public static void viewShipPosition ()

        {

            Console.WriteLine("Enter Ship Serial Number To Find Its Position:");

            string serialNum = Console.ReadLine();

            ship req = shipDL.findShip(serialNum);

            if (req != null)

            {

                shipUI.printLocation(req.getShipLon() , req.getShipLat());

            }

            else

            {

                Console.WriteLine("Not Exist");

            }

        }

        public static void changePosition ()

        {

            Console.WriteLine("Enter Ship Serial Number To Change Its Position:");

            string serialNum = Console.ReadLine();

            ship req = shipDL.findShip(serialNum);

            if (req != null)

            {

                Console.WriteLine("Enter Latitude's Degree:");

                int degree = int.Parse(Console.ReadLine());

                Console.WriteLine("Enter Latitude's Minutes:");

                float minutes = float.Parse(Console.ReadLine());

                Console.WriteLine("Enter Latitude's Direction:");

                char direction = char.Parse(Console.ReadLine());

                angle lat = req.getShipLat();

                lat.changePosition(degree , minutes , direction);

                Console.WriteLine("Enter Ship Longitude:");

                Console.WriteLine("Enter Longitude's Degree:");

                degree = int.Parse(Console.ReadLine());

                Console.WriteLine("Enter Longitude's Minutes:");

                minutes = float.Parse(Console.ReadLine());

                Console.WriteLine("Enter Longitude's Direction:");

                direction = char.Parse(Console.ReadLine());

                angle lon = req.getShipLon();

                lon.changePosition(degree , minutes , direction);

            }

            else

            {

                Console.WriteLine("Not Exist");

            }

        }

        public static void viewShipSerialNum ()

        {

            Console.WriteLine("Enter Ship Latitude:");

            string latitude = Console.ReadLine();

            Console.WriteLine("Enter Ship Longitude:");

            string longitude = Console.ReadLine();

            string shipName = shipDL.findShipName(latitude , longitude);

            if (shipName != null)

            {

                Console.WriteLine(shipName);

            }

            else

            {

                Console.WriteLine("Not Exist");

            }

        }

    }

**UI Code: angleUI.cs**

class angleUI

    {

        public static void displayPosition (angle myAngle)

        {

            Console.WriteLine($"{myAngle.getDegrees()}\u00b0{myAngle.getMinutes()}' {myAngle.getDirection()}");

        }

        public static string displayStringFormat (angle myAngle)

        {

            string x = $"{myAngle.getDegrees()}'{myAngle.getMinutes()}' {myAngle.getDirection()}";

            return x;

        }

    }

**UI Code: MenuUI.cs** **Driver Programs: Program.cs**

class MenuUI

   {

       public static int mainMenu ()

       {

           int op;

           Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

           Console.WriteLine("       OCEAN NAVIGATION       ");

           Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

           Console.WriteLine("1- Add Ship");

           Console.WriteLine("2- View Ship Position");

           Console.WriteLine("3- View Ship Serial Number");

           Console.WriteLine("4- Change Ship Position");

           Console.WriteLine("5- Exit");

           op = int.Parse(Console.ReadLine());

           return op;

       }

       public static void clearScreen ()

       {

           Console.WriteLine("Press any key to continue...");

           Console.ReadKey();

           Console.Clear();

       }

   }

class Program

    {

        static void Main ()

        {

            string path = "ship.txt";

            shipDL.loadFromFile(path);

            int op = 0;

            while (op < 5)

            {

                MenuUI.clearScreen();

                op = MenuUI.mainMenu();

                if (op == 1)

                {

                    MenuUI.clearScreen();

                    ship newShip = shipUI.addShip();

                    shipDL.addShipIntoList(newShip);

                    shipDL.storeIntoFile(path , newShip);

                }

                else if (op == 2)

                {

                    MenuUI.clearScreen();

                    shipUI.viewShipPosition();

                }

                else if (op == 3)

                {

                    MenuUI.clearScreen();

                    shipUI.viewShipSerialNum();

                }

                else if (op == 4)

                {

                    MenuUI.clearScreen();

                    shipUI.changePosition();

                }

            }

        }

    }