namespace Csspl.App.Trigonometry

{

using Csspl.View.Trigonometry.Main;

using Csspl.Process.Trigonometry;

using Csspl.Utils.IO;

public class App

{

private Process process;

public App(Process process)

{

this.process = process;

}

public bool Run()

{

bool running = true;

int startOption = 0;

while (running)

{

Console.Clear();

MainUI.StartView();

startOption = InputHandler.Number("\t\t\t\t> ");

switch (startOption)

{

case 1:

Calculator();

break;

case 2: // Exit

MainUI.YesNoView("Exit the Program?");

running = !InputHandler.YesNoOption("\t\t\t\t> ");

break;

default:

Console.WriteLine("Invalid option. Please try again.");

break;

}

InputHandler.ContinueKey();

}

Console.WriteLine("\t\t\t\tGoodbye :)\n");

return running;

}

private void Calculator()

{

int inputOption = 0;

bool state = true;

while (state)

{

InputHandler.ContinueKey();

Console.Clear();

MainUI.CalculationView(); // Display the options for calculations

inputOption = InputHandler.Number("\t\t\t\t> ");

switch (inputOption)

{

case 0:

MainUI.YesNoView("Return to Main Menu?");

state = !InputHandler.YesNoOption("\t\t\t\t> ");

if(state) continue;

else return;

case 1:

process.ComputeOppositeUsingSine();

break;

case 2:

process.ComputeOppositeUsingTan();

break;

case 3:

process.ComputeHypotenuseUsingSine();

break;

case 4:

process.ComputeHypotenuseUsingCosine();

break;

case 5:

process.ComputeAdjacentUsingCosine();

break;

case 6:

process.ComputeAdjacentUsingTan();

break;

case 7:

process.ComputeAngleUsingSine();

break;

case 8:

process.ComputeAngleUsingCosine();

break;

case 9:

process.ComputeAngleUsingTan();

break;

case 10:

process.ComputeSidesOnRadian();

break;

case 11:

process.ComputeRadianOnAngle();

break;

default:

Console.WriteLine("Invalid option.");

break;

}

if(inputOption != 0 && (inputOption >= 1 && inputOption <= 11))

MainUI.ResultView(process.Result);

}

}

} /\* class App \*/

}/\* namespace Csspl.App.Trigonometry \*/