To implement a program that handles different shapes (Circle, Rectangle, Triangle) and calculates their areas using TypeScript, we need to follow these steps:

1. Define the types for Circle, Rectangle, and Triangle.
2. Define a union type Shape that encompasses these types.
3. Implement a function calculateArea that uses type guards to determine the shape type and calculates the area accordingly.

Here's how you can do this:

typescript

// Define types for Circle, Rectangle, and Triangle

type Circle = {

kind: 'circle';

radius: number;

};

type Rectangle = {

kind: 'rectangle';

width: number;

height: number;

};

type Triangle = {

kind: 'triangle';

base: number;

height: number;

};

// Define a union type Shape for Circle, Rectangle, and Triangle

type Shape = Circle | Rectangle | Triangle;

// Function to calculate area based on the shape type

function calculateArea(shape: Shape): number {

switch (shape.kind) {

case 'circle':

return Math.PI \* shape.radius \* shape.radius;

case 'rectangle':

return shape.width \* shape.height;

case 'triangle':

return 0.5 \* shape.base \* shape.height;

default:

// This should never happen, provided exhaustive checks

throw new Error("Unsupported shape");

}

}

// Example usage:

const myCircle: Circle = { kind: 'circle', radius: 10 };

const myRectangle: Rectangle = { kind: 'rectangle', width: 5, height: 10 };

const myTriangle: Triangle = { kind: 'triangle', base: 5, height: 8 };

console.log(`Area of myCircle: ${calculateArea(myCircle)}`);

console.log(`Area of myRectangle: ${calculateArea(myRectangle)}`);

console.log(`Area of myTriangle: ${calculateArea(myTriangle)}`);

**Explanation:**

1. **Type Definitions**:
   * Circle, Rectangle, and Triangle are defined with a kind property to distinguish them.
   * Shape is a union type that can be either a Circle, Rectangle, or Triangle.
2. **Type Guards**:
   * The calculateArea function uses a switch statement to check the kind property of the Shape type.
   * Depending on the value of kind, it calculates the area using the appropriate formula.
3. **Usage**:
   * Example objects for Circle, Rectangle, and Triangle are created.
   * The calculateArea function is called with each of these objects, and the areas are logged to the console.