TypeScript to Solidity Compiler

Project Overview

---------------

A compiler that transforms TypeScript classes into Solidity smart contracts, enabling developers to write smart contracts using TypeScript syntax and features.

Core Features

------------

- TypeScript class to Solidity contract conversion

- Support for class inheritance and interfaces

- Automatic generation of Solidity pragma directives

- Output compilation to .sol files

Technical Architecture

--------------------

- Built with TypeScript

- Uses ts-morph for TypeScript AST manipulation

- Implements custom class parser for Solidity code generation

- Supports modern Solidity version (^0.8.0)

Project Structure

----------------

contracts/ - Source TypeScript contract files

compiled/ - Generated Solidity files

parsers/ - Custom parsers for code transformation

specs/ - Project documentation

Development Requirements

----------------------

- Node.js environment

- TypeScript

- ts-morph library

- reflect-metadata for TypeScript decorators

Usage

-----

1. Write TypeScript classes in the contracts directory

2. Run the compiler

3. Generated Solidity files will be available in the compiled directory

Future Enhancements

------------------

- Support for more Solidity features

- TypeScript decorator support

- Contract testing framework

- Gas optimization features