2-2: Control Flow and Pattern Matching (Practice)

Artem Pavlov, TII, Abu Dhabi, 04.05.2024

Create new crate

- Create new branch in the repository p22
- Create new library crate p22
- Check that p22 is listed as a member of the workspace in the root Cargo.toml

Calculator

- Create calc module
- Write 3 functions in it:
  - celsius2farenheit(celsius: i32) -> i32
  - farenheit2celsius(farenheit: i32) -> i32
  - fibonacci_loop(n: u32) -> u64 (with loop, without using recursion)
  - fibonacci_rec(n: u32) -> u64 (with match and recursion)
- Write doctests, unit tests, and integration tests for them
- Write benchmarks for the Fibonacci functions

Song

- Create song module
- Create a function which prints the lyrics to the Christmas carol “The Twelve Days of Christmas,” taking advantage of the repetition in the song.
- Create a binary (i.e. file in src/bin/) which executes this function

Figures

- Create figures module
- Define the following types:
  - Point: contains x and y fields
  - Circle: contains center point and radius
  - Triangle: contains 3 points
  - Rectangle: contains 2 points
  - Shape: enumeration of point, circle, triangle, and rectangle
- Derive appropriate traits for the types
- Define functions which compute area and perimeter of each type (i.e. you need 5 functions)
- Add tests for each function

Tic-tac-toe

- Create tictac module
- Define TicTacField struct using arrays and enums
- Write analyze function which analyzes the field and can return 4 results: WinX, WinY, WinBoth, GameOn
- Write functions which modifies the field with the following signature:

  fn make_move(field: TicTacField, x: u32, y: u32, player: Player) -> Result<TicTacField, Error>
- The Error type should handle possible error cases
- Write tests for both functions