Homework for Error Handling

Odd-Harald Lillestø Myhren

July 24, 2023

Understanding the Result enum

1. Define a function `divide_numbers` that takes two `i32` integers as input and returns a result of their division. If the second number is zero, return an `Err` variant with the message "Division by zero is not allowed."
2. Write a sample code to call the ‘calculate‘ function with different inputs, including a scenario where division by zero occurs. Handle and print the results or error messages accordingly.

Using the ? operator

1. Create a new struct called `Person` with two fields: `name` (String) and `age` (u8).
2. Implement a function named `create_person` that takes two parameters: `name` (String) and `age` (u8). This function should return a `Result` with a `Person` instance as the success value and an error message as the error value if the `name` is empty or the `age` is greater than 120.

The panic! macro

1. Write a function called `find_element` that takes a vector of integers and an index (usize) as parameters. The function should return the value at the given index if the index is within the vector's bounds. If the index is out of bounds, use the `panic!` macro to generate a suitable error message.
2. Create a vector of integers with some arbitrary values and call the `find_element` function with different index values. Observe how the `panic!` macro behaves and how it affects the program execution.