Лабораторна робота 3. Робота з файлами

Зчитування з файлу:

```
use std::fs::File;
use std::io;

use std::io::prelude::*;
struct Lines<R> {
    reader: io::BufReader<R>,
    buf: String
}
impl <R: Read> Lines<R> {
    fn new(r: R) -> Lines<R> {
        Lines { reader: io::BufReader::new(r), buf: String::new() }
    }
    fn next<'a>(&'a mut self) -> Option<io::Result<&'a str>>{
        self.buf.clear();
        match self.reader.read_line(&mut self.buf) {
            Ok(nbytes) => if nbytes == 0 {
                None // no more lines!
            } else {
                let line = self.buf.trim_right();
                Some(Ok(line))
            },
            Err(e) => Some(Err(e))
        }
        }
}
fn read_all_lines(filename: &str) -> io::Result<()> {
    let file = File::open(&filename)?;

    let mut lines = Lines::new(file);
    while let Some(line) = lines.next() {
        let line = line?;
        println!("{}", line);
    }

    Ok(())
}
```

Запись у файл:

```
use std::fs::File;
use std::io;
use std::io::prelude::*;

fn write_out(f: &str) -> io::Result<()> {
    let mut out = File::create(f)?;
    write!(out,"answer is {}\n", 42)?;
    Ok(())
}

fn main() {
    write_out("test.txt").expect("write failed");
}
```

Відомості про файл (його розмір, тип тощо)

```
use std::env;
use std::path::Path;

fn main() {
    let file = env::args().skip(1).next().unwrap_or("test.txt".to_string());
    let path = Path::new(&file);
```

```
    match path.metadata() {
        Ok(data) => {
            println!("type {:?}", data.file_type());
            println!("len {}", data.len());
            println!("perm {:?}", data.permissions());
            println!("modified {:?}", data.modified());
        },
        Err(e) => println!("error {:?}", e)
    }
}
```

Щоб знайти вміст каталогу використовуємо ітератор. Ось усі файли з розширенням '.rs' і розміром понад 1024 байт

```
use std::fs;
use std::io;
fn dump_dir(dir: &str) -> io::Result<()> {
    for entry in fs::read_dir(dir)? {
        let entry = entry?;
        let data = entry.metadata()?;
        let path = entry.path();
        if data.is_file() {
            if let Some(ex) = path.extension() {
                if ex == "rs" && data.len() > 1024 {
                    println!("{} length {}", path.display(),data.len());
                }
            }
        }
    }
    Ok(())
}

fn main() {
    dump_dir("src");
}
```

Завдання:

Напишіть програму, яка виконує функції файлового менеджера. Необхідні дії:

- ✓ створення нового текстового файлу,
- ✓ створення нового каталогу,
- ✓ копіювання файлу,
- ✓ скопіювання каталогу (разом із файлами),
- ✓ пошук,
- ✓ видалення 1 файлу на ім'я,
- ✓ видалення каталогу,
- ✓ видалення декількох каталогів,
- ✓ перегляд вмісту каталогу,
- ✓ перегляд властивостей файлу,
- ✓ перегляд властивостей каталогу.