Języki programowania i GUI

Lista 1 - 2025

1. Odwiedź stronę `https://www.rust-lang.org/tools/install` o zainstaluj `rustup`. Zainstaluj Visual Studio Code i wtyczkę `rust-analyzer`. Następnie utwórz pierwszy projekt i uruchom go: `cargo new zad1; cd zad1;cargo run;`. Otwórz folder `zad1` w vscode: `code zad1` i zacznij pracę. W podobny sposób twórz środowiska pracy do następnych zadań.

2. Rusta dobrze się uczyć ze strony `https://doc.rust-lang.org/book/`. Przeczytaj rozdziały 1.3, 3.1, 3.2 .. . 3.5. Następnie napisz funkcję `nwd(mut a:i32,mut b:32)->i32` obliczającą największy wspólny dzielnik liczb a i b, oraz funkcję `test_nwd`, która będzie sprawdzała poprawność jej działania za pomocą kilku wywołań makra `assert_eq!` Nad funkcją `test_nwd` umieść anotację `#[test]` i przetestuj program poleceniem `cargo test`.

3. Napisz program, który wyprodukuje ładnie sformatowaną table wartość funkcji trygonometrycznych w postaci tabeli zawierających kolumny: kąt, sinus, cosinus, tangens. dla wartości kątów od 0^o do 45^o.
   Wskazówka: `println!("| {x:2} | {s:5} | {c:5} | {t:5} |")`.

4. Utwórz projekt `zad3` i wykonaj w nim wszystkie kroki opisane w rozdziale *2. Programming a Guessing Game*. Zagraj w nią kilka razy. Jaka ilość zgadnięć gwarantuje wygraną?

5. Następnie wykonaj podobną grę wg własnego pomysłu. Na przykład wordle.

6. Przeczytaj rozdziały 4.1, 4.2 i 4.3. Następnie napisz i przetestuj następujące funkcje:

   (a) `fn sum(t: &[i32])->i32` suma elementów tablicy `t`.

   (b) `fn max_ascend(t: &[i32])->&[i32]` najdłuższy podciąg rosnący tablicy `t`

   (c) `fn accumulate(t: &mut [i32])` każdy element tablicy powiększa o sumę poprzednich

7. Przeczytaj rozdział 5. Zaprojektuj strukturę opisującą prostokąt umiejscowiony w układzie współrzędnych (oprócz wymiarów należy pamiętać współrzędne środka lub któregoś z wierzchołków). Zaimplementuj metody obliczające jego: pole, obwód oraz metody wykonujące operacje na nim: przesunięcie o wektor, przesuniecie do punktu, obrót o 90^o oraz przeskalowanie o podany współczynnik. napisz zbiór funkcji testujących całą implementację i sprawdź czy wszystko dobrze działa uruchamiając `cargo test`.