

Języki programowania i GUI

Lista 3 - 2025

1. Uogólnij funkcję `eval` w klasie `Poly`, na funkcję uogólnioną (szablon funkcji) `eval<T>`, która będzie działać poprawnie dla każdego `T`, które implementuje traity `Add`, `Sub`, `Mul` oraz `Mul<f64>`, `Add<f64>`, `Sub<f64>`. Wynikiem ma być obiekt klasy `T`. Po prawidłowym zaimplementowaniu, powinno być możliwe obliczenie współczynników wielomianu $P(Q(x))$ za pomocą instrukcji `p.eval(q)`, gdzie `p` i `q` są wielomianami.

Po niewielkich modyfikacjach (i dodaniu traitu `From<f64>`) twoja funkcja powinna też móc wyliczać wartość wielomianu dla argumentów będących macierzami kwadratowymi, ale to nie wchodzi w zakres zadania.
 2. Użyj biblioteki `rug` by znaleźć 10_000-ną cyfrę liczby `Fib(1_000_000)`. Możesz użyć implementacji podanej na wykładzie 3.
 3. Zaimplementuj strukturę `Osoba` zawierającą jej imię, nazwisko, wzrost, masę ciała i datę urodzenia (może być w postaci `rrrr-dd-mm`). Utwórz wektor zawierający dane kilkunastu osób. Następnie napisz funkcje które:
 - (a) posortują osoby alfabetycznie (imię, nazwisko). Użyj `sort_by_key`.
 - (b) posortują osoby wg wieku, Użyj `sort_by_key`.
 - (c) zwracającą nowy wektor referencji do osób, która mają wzrost w przedziale `(a,b)`.
 - (d) obliczającą łączną masę oraz średnią masę wszystkich osób.
 - (e) obliczającą sumę oraz średnią dowolnej funkcji anonimowej której argumentem jest `&Fig` a wynikiem liczba `f64`.
 4.
 - (a) Naucz się stosować bibliotekę `serde` tak, by można było przechowywać spis osób w pliku `json` na dysku.
 - (b) Napisz program działający w konsoli, który pozwoli wykonywać na tej liście następujące operacje: `list_all`, `add`, `remove`, `find`. Po dodaniu lub usunięciu osoby z listy, plik na dysku powinien być automatycznie zapisywany. Lista osób powinna być ładowana z pliku w momencie uruchomienia aplikacji. Procedura `find` powinna prosić o podanie fragmentu nazwiska i wyświetlać osoby o pasujących nazwiskach.
 - 4'. Alternatywnie można poprzednie zadanie wykonać z wykorzystaniem bazy danych `sqlite3` lub `mysql` zamiast biblioteki `serde`.
 5. Zrealizuj funkcje `load` i `save` dla wektora figur z polimorfizmem zadany przy pomocy traitów.
 6. Dodaj do programu wizualizację figur za pomocą biblioteki `flo`. Rozszerz w miarę potrzeby trait `Fig` o funkcję `paint` rysującą figurę na ekranie.
 7. * (Projekt) Wzorując się (lub nie) na przykładzie z wykładu stwórz przy pomocy biblioteki `rocket` i bazy danych `mysql` serwis z newsami.
 8. * (projekt) Zrealizuj pomysł wizualizacji symulacji fizycznej lub jakiejś gry typu strzelanka, platformowa lub planszowa za pomocą biblioteki `flo` lub innej biblioteki graficznej. Zacznij od zaprojektowania struktury, w której będziesz przechowywać stan gry.
- * Z tych zadań wybieramy jedno, uzgadniamy z prowadzącym pomysł i założenia i realizujemy je w czasie do połowy semestru. Można też wystąpić z własnym pomysłem aplikacji i go uzgodnić.