

Języki programowania i GUI

Przygotowanie do kolokwium z Rusta

1. Wypisz oddzielone spacjami liczby nieparzyste od 99, do 1 używając:
 - (a) pętli `while`,
 - (b) pętli `loop`,
 - (c) pętli `for`.
2. Za pomocą jednej instrukcji `let` utwórz zmienne `imie`, `wzrost`, `waga` nadając im wartości `"Jan"`, `180`, `77.5`.
3. Stwórz zmienną `ones`, która będzie tablicą zawierającą 100 jedynek typu `i32`.
4. Stwórz zmienną `box_of_ones`, która będzie tablicą zawierającą 100 takich tablic.
5. Zapisz w zmiennej `c` większą z liczb `a`, `b` za pomocą: (a) instrukcji `if` (b) wyrażenia `if`
6. Napisz zagnieżdżone wyrażenie `if` zwracające największą z trzech liczb: `a`, `b`, `c`.
7. Za pomocą pętli `for` z tablicy `tab:Vec<i32>` wypisz elementy ujemne rozdzielone spacjami.
8. Napisz wyrażenie `match`, którego wartością będzie liczba dni miesiąca `m` ($m \in 1..12$) w roku nieprzestępnym.
9. W funkcji `fn policz(tab: &Vec<i32>)->(i32,i32)` w jednej pętli `for` policz, ile elementów tablicy: `tab` jest ujemnych, a ile dodatnich. Zwróć wynik jako krotkę `(i32,i32)`.
10. Dla struktury `struct vec2D(f64,f64)` napisz wyrażenie typu `vec2D` zwracające sumę wektorów `a`, `b`, które oczywiście też są zmiennymi typu `vec2D`.
11. Stwórz strukturę `Trójkąt` która będzie pamiętała długości `a,b,c` boków trójkąta.
 - (a) Napisz funkcję `obwód`
 - (b) zaimplementuj metodę `obwód`.
 - (c) zaimplementuj `trait Obwód { fn obwód(&self)->f64;}`
12. Dla zmiennej `t` typu `Trójkąt` zastosuj destrukuryzację, by stworzyć zmienne `a`, `b`, `c` będące długościami jego boków i sprawdź czy spełniają one warunek trójkąta zwracając `true` lub `false`. Pamiętaj, że nie wiadomo z góry, który bok jest najdłuższy.
13. Dla struktury:

```
enum Trójmian{
    Ogólna{a:f64,b:f64,c:f64},
    Kanoniczna{a:f64,p:f64,q:f64},
    Iloczynowa{a:f64,x1:f64,x2:f64},
}
```

 - (a) Napisz metodę `fn eval(&self,x:f64)->f64` która zwróci wartość trójmianu dla podanego argumentu `x`.
 - (b) Napisz metodę `fn na_ogólną(&self)->Trójmian`, która zwróci postać ogólną Trójmianu `f`. Użyj `match` tak, aby metoda działała prawidłowo dla każdego z trzech wariantów struktury `Trójmian`.

Wskazówka: $f(x) = ax^2 + bx + c = a(x - p)^2 + q = a(x - x_1)(x - x_2)$. Postać ogólną otrzymujemy z iloczynowej lub kanonicznej przez wymnożenie.