

Języki programowania i GUI

Lista 6 - 2023

1. W Julii makra działają na całkowicie innej zasadzie niż w C++. Makro jest po prostu funkcją, która jest uruchamiana w trakcie kompilacji programu a jej zadaniem jest generowanie kodu programu, który powinien być wynikiem makra. Wewnątrz makr można używać wszystkich konstrukcji języka Julia, w tym pętli, oraz funkcji. Jedy-
nym ograniczeniem jest fakt, że wynikiem makra musi być kod programu w postaci
wyrażenia. Na przykład:

```
macro pokaż(x)  
    :(print("$x="), $x)  
end  
y=45  
@pokaż(y)      # y=45  
@pokaż(3+y/5)  #2 + y / 5=11.0
```


Napisz makro `twice` tak, aby instrukcja `@twice(czynność)` wykonywała czynność dwu-
krotnie. np `@twice(println("hello"))`.
2. (2pkt) Napisz funkcję `simpler`, która będzie wprowadzać wyrażenia algebraiczne typu `Expr` do prostszej postaci. Wymagania:
 - funkcja powinna najpierw rekurencyjnie upraszczać podwyrażenia
 - liczby i symbole pozostają nie zmienione: `simpler(a::Union{Number,Symbol})=a`
 - wyrażenia, których argumentami są wyłącznie liczby, zastępowane są wyliczoną wartością np `:(1 + 2 + 4) => 7`
 - z iloczynów usuwane są jedynki a z sum zera `:(x + 0 + y) => :(x + y)`
 - jeśli w iloczynie występuje 0 wynik jest zastępowany zerem `:(x * y * 0) => 0`
 - Jeśli na skutek uproszczeń w iloczynie lub sumie został już tylko jeden argument to jest on wynikiem `:(x + 0) -> :(+ x) => :x`
 - * liczby występujące w sumach siebie do siebie dodawane a w iloczynach mnożone `:(x + 3 + y + 5) => :(x + y + 8)` oraz `:(x * 3 * y * 5) => :(x * y * 15)`
3. (3pkt) Napisz funkcję `pochodna`, która będzie wyliczać pochodną wyrażenia (`Expr`) względem symbolu `:x`. Wymagania:
 - pochodna liczby jest zerem: `pochodna(a::Number)=0`
 - pochodna symbolu różnego od `x` jest zerem, a pochodna `:x` wynosi 1
 - pochodna sumy jest sumą pochodnych. Podobnie dla różnicy.
 - prawidłowo wyliczane są pochodne iloczynu oraz ilorazu
 - prawidłowo wyliczana jest pochodna funkcji złożonej
 - prawidłowo wyliczane są pochodne potęg (np. `x^5`), oraz `5^x`
 - prawidłowo wyliczane są pochodne funkcji `sin`, `cos`, `tan`, `log`, `exp`, `sqrt`

Napisz program testujący, który w pętli prosi użytkownika o podanie wzoru funkcji. Następnie drukuje wylicza jego pochodną i wynik upraszcza funkcją z poprzedniego zadania.
4. (2pkt) Używając pakietu np. `Plots` wykonaj wykres kilka przykładowych wykresów na jednym rysunku. Zastosuj argumenty kluczowe: `legend`, `linewidth`, `title`... Następnie napisz program, który prosi użytkownika o podanie wzoru funkcji (i ewentualnie zakresu zmiennej `x`) następnie rysuje wykres tej funkcji (oraz ewentualnie jej pochodnej). Uwaga: potrzebne będzie użycie `Meta.parse`, `eval`, a także `Base.invokelatest`.