

Zadanie 2.1

Uruchom interpreter języka Haskell (polecenie **ghci**). Wyznacz wartości następujących wyrażeń (zapisz je zgodnie ze składnią języka Haskell):

1. 2^{100}
2. $(12 + 23)(40 - 21)$
3. iloraz pierwszego z wyrażeń przez drugie
4. pierwiastek kwadratowy z 2
5. pierwiastek kwadratowy z $(12 + 23)(40 - 21)$
6. $2 + -3$
7. $2 + (-3)$

Zadanie 2.2

1. Korzystając z funkcji **rem** sprawdź, które z liczb z 2, 3, 4 są dzielnikami liczby 1234.
2. (*) Korzystając z funkcji **rem** sprawdź, czy 113 jest liczbą pierwszą.

Zadanie 2.3

Korzystając z polecenia **let** zdefiniuj funkcje wyznaczające:

1. objętość kuli (**sphereVolume**);
2. objętość stożka (**coneVolume**);
3. (*) objętość prostopadłościanu (**cuboidVolume**).

Przetestuj działanie tych funkcji na przykładowych danych.

Zadanie 2.4

Zdefiniuj moduł **Solids** (zapisz go w pliku **solids.hs**) i umieść w nim definicje funkcji z zadania 2.3. Do każdej funkcji dodaj informację o jej typie oraz komentarz opisujący jej działanie. Załaduj moduł **Solids** do interpretera (polecenie **:load** lub **:l**) i przetestuj działanie poszczególnych funkcji.

Zadanie 2.5

(*) Do modułu **Solids** dopisz funkcję wyznaczającą objętość ostrosłupa o podstawie kwadratowej (**pyramidVolume**). Przetestuj działanie tej funkcji.

Zadanie 2.6

Zdefiniuj moduł **Math** i umieść w nim definicje funkcji **f1**,...,**f8** określonych jak poniżej. Zdefiniuj funkcje w dwóch wersjach: tradycyjnej i używając sekcji.

1. **f1** zmniejsza pobierany argument o 2; **Uwaga:** $(+(-2))$
2. **f2** zwraca resztę z dzielenia argumentu przez 5;
3. **f3** zwraca wynik porównania argumentu z liczbą 10;
4. **f4** zwraca część całkowitą z dzielenia argumentu przez 8;
5. (*) **f5** zwraca **True**, jeżeli argument jest większy od 100 i **False** w przeciwnym przypadku;
6. (*) **f6** zwraca resztę z dzielenia liczby 5 przez argument funkcji;

7. (*) **f7** zwraca część całkowitą z dzielenia liczby 8 przez argument funkcji;
8. (*) **f8** potraja pobierany argument.

Załaduj moduł **Math** i przetestuj działanie tych funkcji.

Zadanie 2.7

(*) Zdefiniuj moduł **Triangle** i umieść w nim definicje funkcji **trArea1**, **trArea2** i **trArea3**, które wyznaczają pole powierzchni trójkąta, przy czym każda z tych funkcji wykorzystuje inny wzór na pole trójkąta.