

1. Przerobić kod z podstaw arytmetyki wskaźnikowej na taki w którym stosujemy std::unique_ptr

Napisz funkcję smartZadanie1().

Funkcja ta ma być lustrzanym odbiciem funkcji zadanie1, ale proszę wykorzystać unique_pointer'a zamiast zwykłego wskaźnika

2. Sharepointery

- 2.1. Stwórz sharepointery dla 3 rodzajów kont
- 2.2. Stwórz vector sharedpointerów
- 2.3. Dodaj każdy z sharedpointerów do tego vectora

3. Weak pointer

- 3.1. napisz klasę Person, która posiada imię oraz wskaźnik na partnera
 - 3.2. napisz konstruktor i destruktor z wypisywaniem na konsole (aby lepiej zobaczyć działanie smart pointerów)
 - 3.3. napisz funkcję Love() która paruje dwa obiekty typu Person tj. jeden obiekt wskazuje na drugi a drugi wskazuje na pierwszy
 - 3.4. odkomentuj zawartość main i zobacz czy poprawnie wykonałeś powyższe zadania
- wskazówka - pamiętaj, aby użyć właściwego rodzaju smart pointera tak aby zapobiec odniesienia cyklicznego (circular reference)

4. Mix

- 4.1. Stwórz funkcję make, która zwróci unique_ptr wektora shared_ptrs obiektu Point
- 4.2. Stwórz funkcję fill, która przyjmuje jako argumenty: vector shared pointerów obiektu Point i wartość int, która reprezentuje, ile zostanie wprowadzonych wartości.

Działanie funkcji:

- 1. Pyta użytkownika, ile liczb chce podać
- 2. Tworzy shared_ptr
- 3. Dodaje ten shared_ptr do vectora