1. Za abstraktni podatkovni tip Int z definiranimi operacijami **succ, pred, add, neg in sub** pokažite, da je: **sub(neg(n), m) = neg(add(n, m))**.

Vemo:

* succ(0) = 1,
* succ(succ(succ(…(0)…))) (n-krat) lahko zapišemo kot succn(0) = n,
* neg(n) = succ-n(0),
* pred(0) = -1,
* pred(pred(pred(…(0)…))) (n-krat) lahko zapišemo kot predn(0) = neg(succn(0))

Dokazujemo resničnost: **sub(neg(n), m) = neg(add(n, m))**

Leva stran:

n … succn(0)  
m … succm(0)

1. sub(neg(n), m)
2. sub(neg(succn(0)), succm(0))
3. sub(succ-n(0), succm(0))
4. succ-(n-m)(0)
5. predn+m(0)

Desna stran:

n … succn(0)  
m … succm(0)

1. neg(add(n, m))
2. neg(add(succn(0), succm(0))
3. neg(succn+m(0))
4. predn+m(0)

Sklep: enakost velja.

2. Napišite javansko implementacijo metode **obrni(Stack s, int n, int m)**, ki obrne m elementov sklada s od mesta n dalje. Rezultat je spremenjen izhodiščni sklad, pri reševanju pa lahko uporabljate pomožne sklade. Za izvedbo sklada uporabite razred Stack iz Collection Framework-a. Rešitev naj vsebuje samo operacije nad skladi.

Text

Description automatically generated