

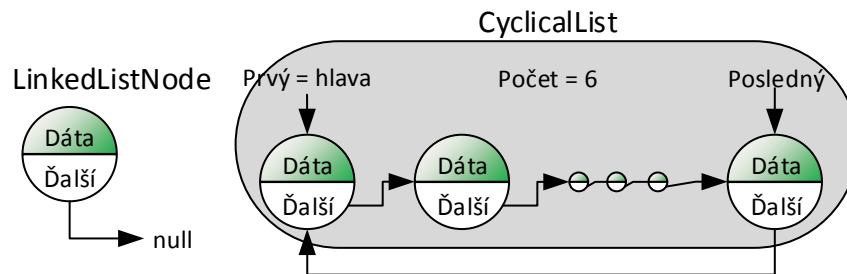
CyclicalList

CyclicalList je implementáciu zoznamu pomocou navzájom zreťazených prvkov (v našom prípade LinkedListNode). Každý prvok má odkaz na ďalší prvok zoznamu a dáta. Posledný prvok v zozname ukazuje na prvý prvok v zozname. Zoznam je teda tvorený reťazou objektov typu LinkedListNode. Atribúty LinkedListu potom sú:

- Odkaz na **prvý** prvok (typu LinkedListNode) v zreťazení – teda hlava zoznamu,
- **Počet prvkov** v zozname.

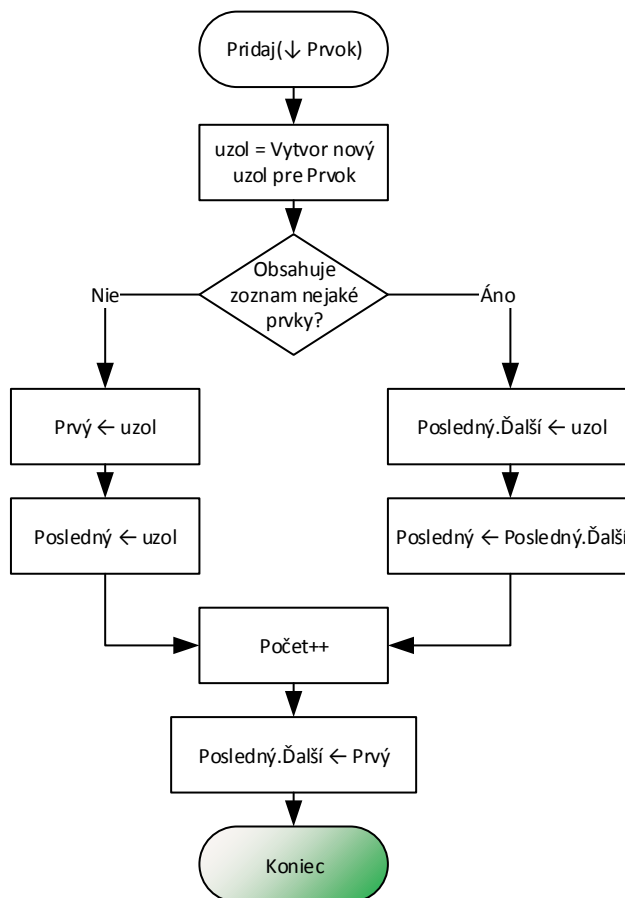
Po vytvorení LinkedList neobsahuje žiadny prvok, teda odkazy na prvý prvok ukazuje na null a počet prvkov je rovný 0.

Každý objekt typu LinkedListNode dokáže sprístupniť a modifikovať svoje položky Dáta a Ďalší. Konštruktor tejto triedy ako parameter preberá Dáta (nemá zmysel vytvárať LinkedListNode, ak nie sú k dispozícii dáta), ktorými sa inicializuje. Odkaz na ďalší prvok je inicializovaný na null.

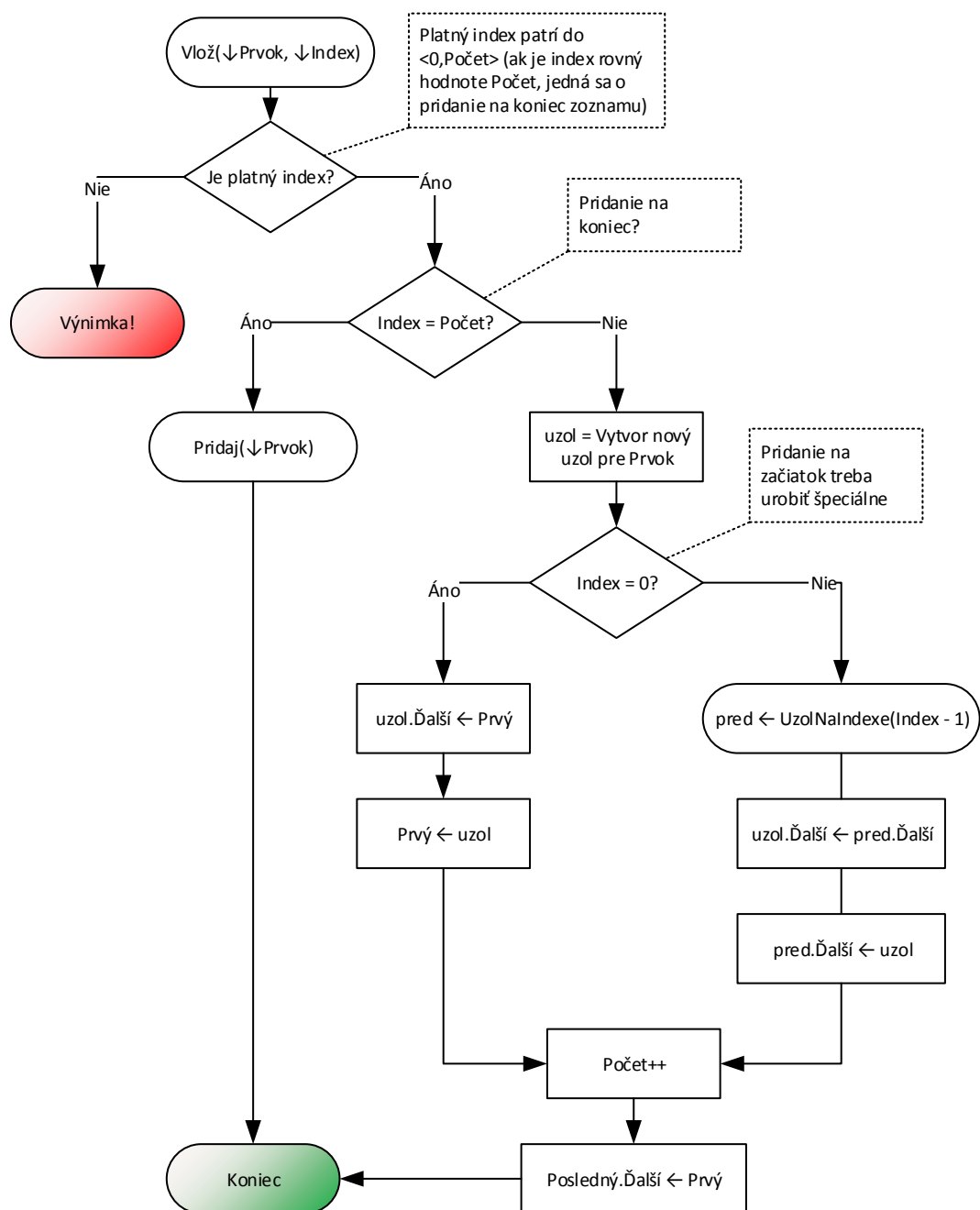


Metódy

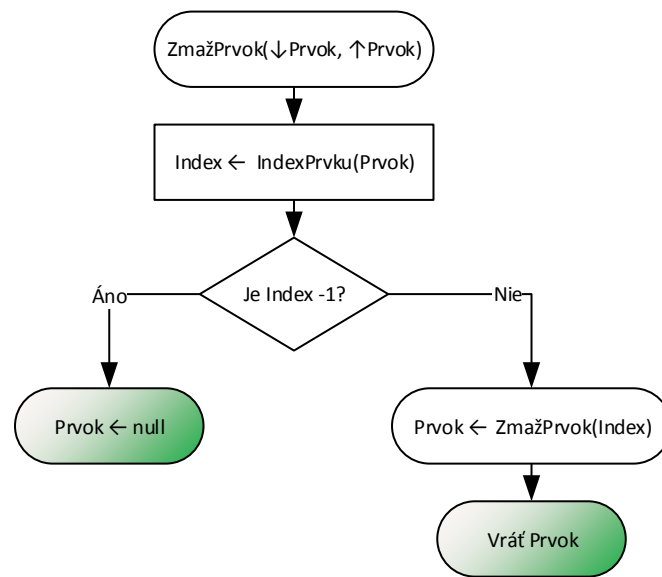
Pridaj prvok pridá prvok na koniec zoznamu (teda pred hlavu).



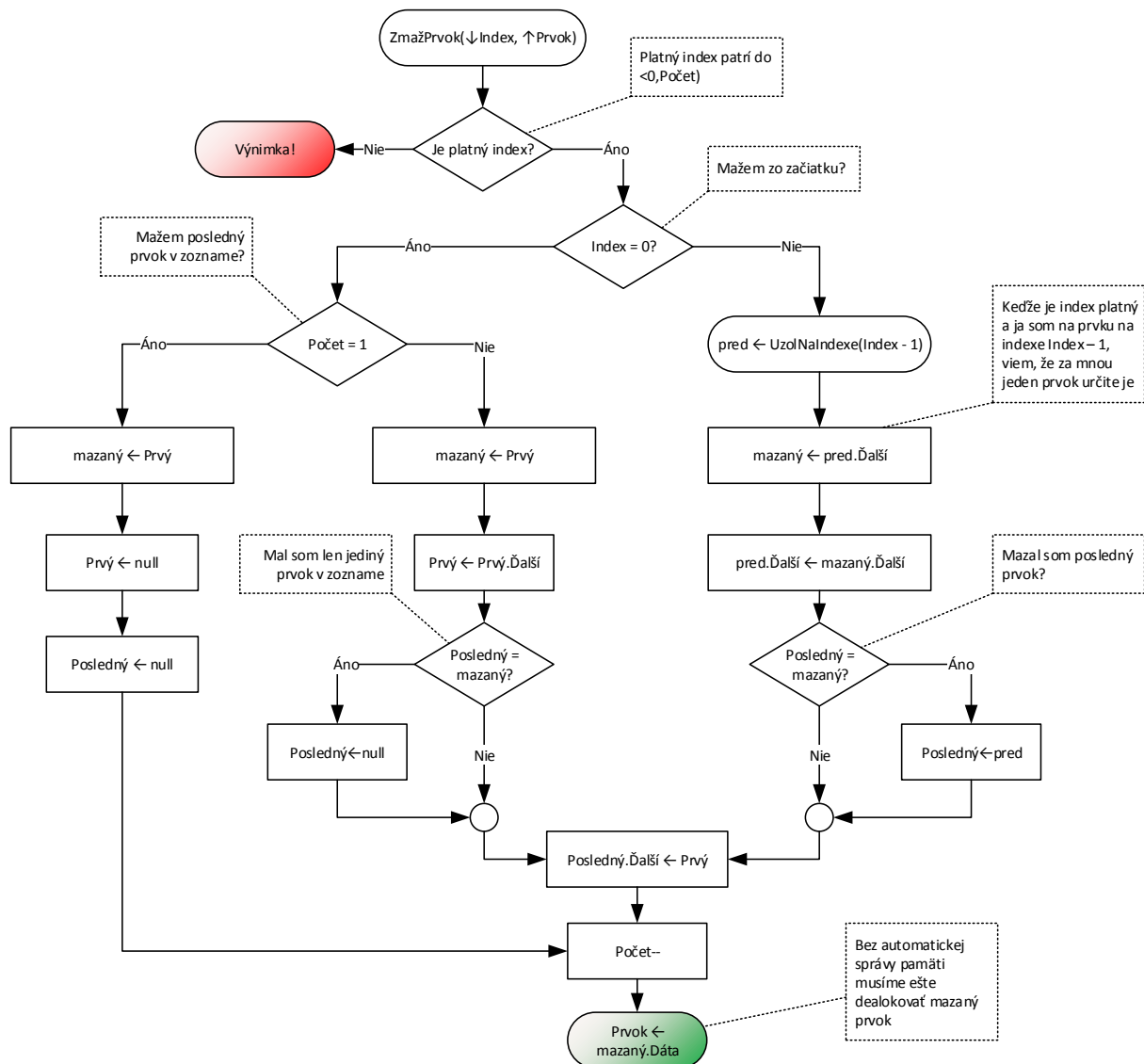
Vlož prvok na daný index vloží (neprepiše) prvok na daný index. Ak je index zhodný s počtom prvkov (teda jedná sa o prvý neplatný index v poli), metóda sa správa ako obyčajné vloženie prvku.



Zmaž prvok nájde prvok v zozname a ten následne vymaže (vymaže sa iba prvý výskyt daného prvku). Na znak úspešného vymazania vráti vymazaný prvok. Ak sa prvok v zozname nenachádza metóda vráti null.

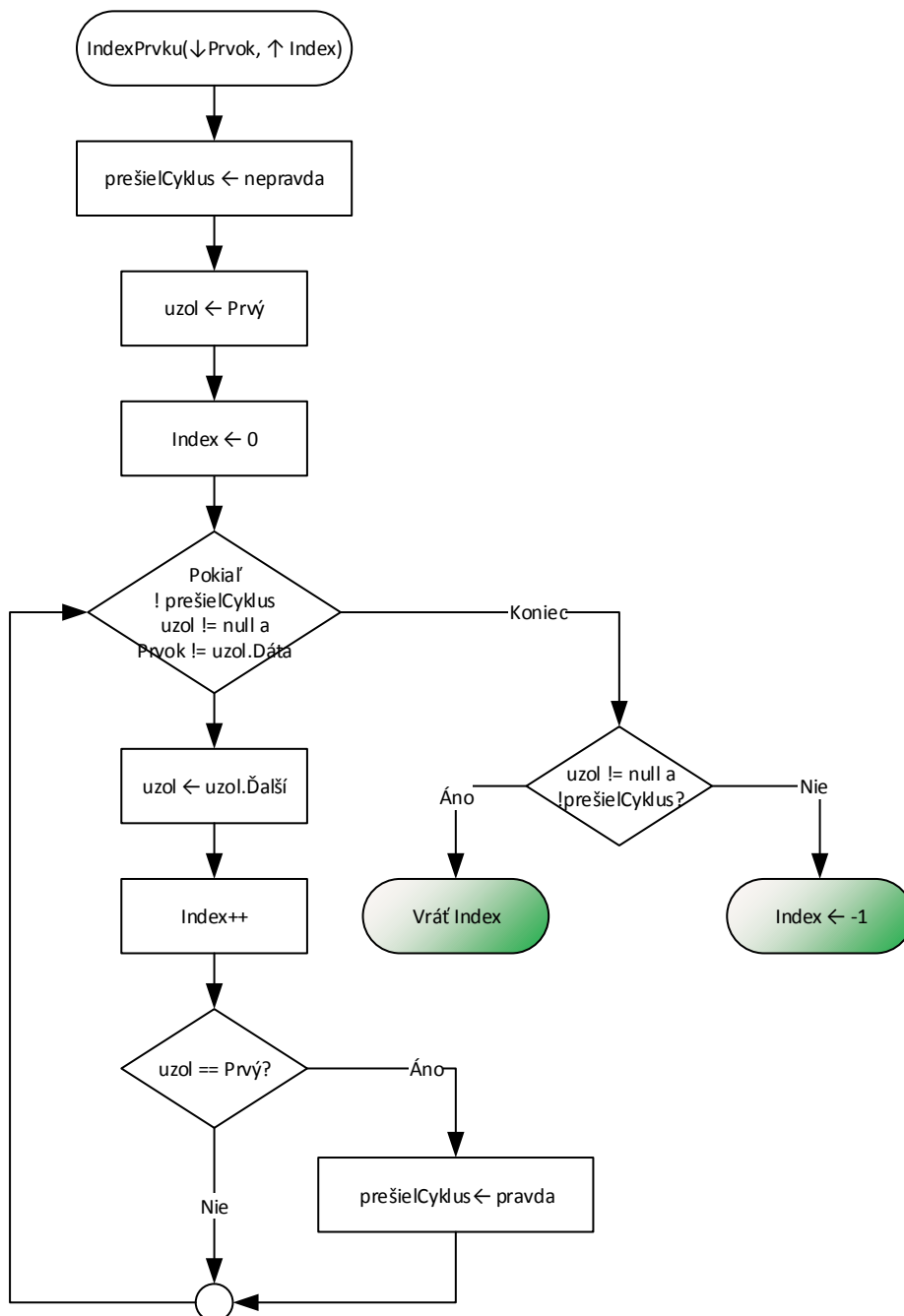


Zmaže prvok na danom indexe (ak nie je index platný, vyhodíme výnimku). Odstránený prvok tvorí návratovú hodnotu tejto metódy (nikdy¹ nad ním nevoláme deštruktor). Naopak, ak nemáme automatickú správu pamäti, nesmieme zabudnúť zavolať deštruktor odstráneného uzla.

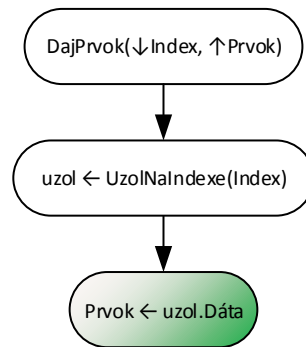


¹ Pokiaľ to nie je požadovaná vlastnosť a očakávané správanie štruktúry.

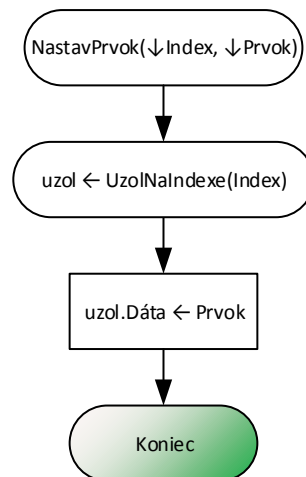
Vráti index prvku, ak sa v zozname nachádza. Vždy vráti index prvej hodnoty (ak sa v zozname nachádza viac rovnakých hodnôt). Referenčné typy porovnávame referenčne²! Ak sa prvok v zozname nenachádza, vráti -1.



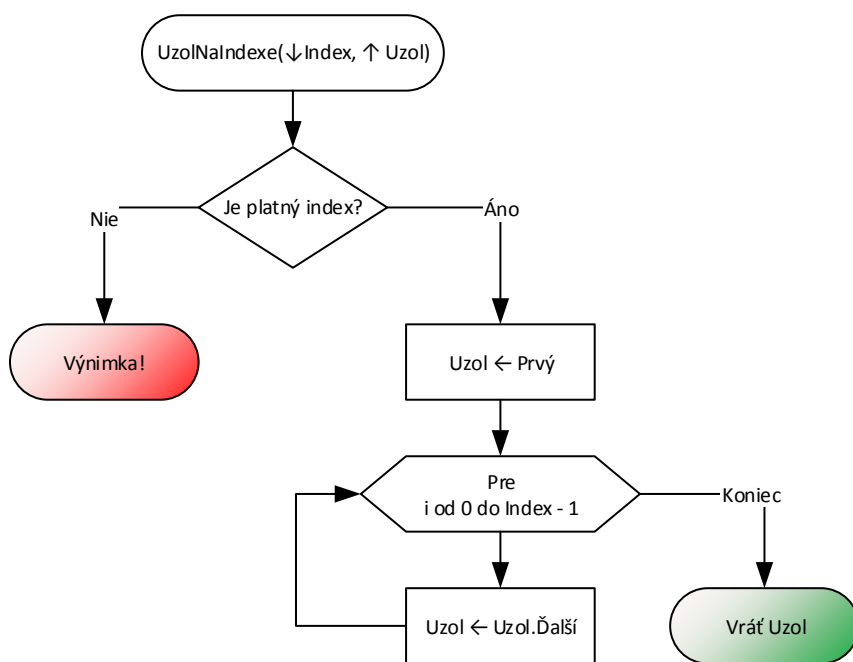
Vráti prvok na danom indexe. O kontrolu správnosti indexu sa stará metóda `UzolNaIndexe`, ktorá v prípade neplatného indexu vystrelí výnimku.



Prepíše prvok na danom indexe. O kontrolu správnosti indexu sa stará metóda `UzolNaIndexe`, ktorá v prípade neplatného indexu vystrelí výnimku.

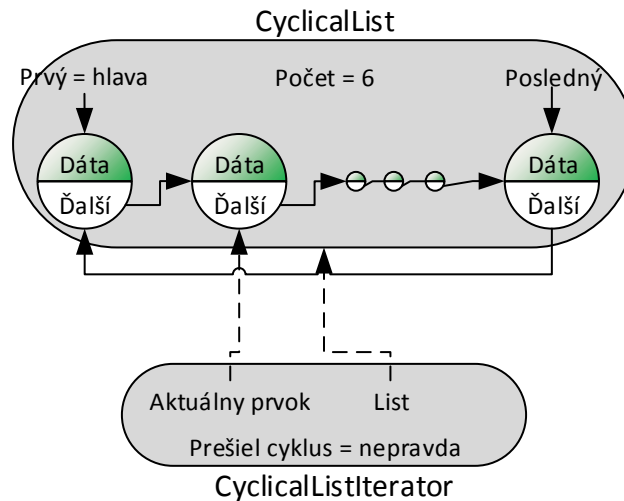


Pomocná metóda, ktorá **vráti uzol na zadanom indexe**. Ak je index neplatný, metóda vystrelí výnimku.

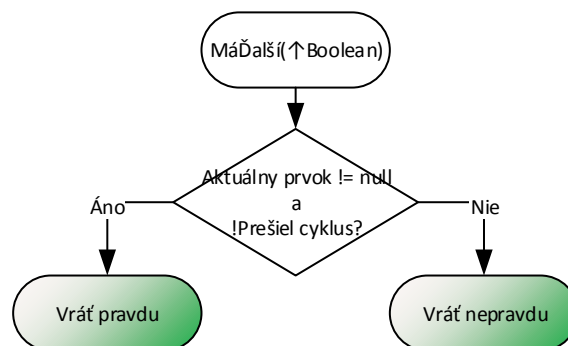


Iterátor

Iterátor pre CyclicalList sa v konštruktore naviaže na zoznam, cez ktorý má iterovať. Obsahuje atribút Aktuálny prvok typu LinkedListNode, ktorý sa inicializuje na hlavu zoznamu. Ďalej si iterátor eviduje, či urobil plnú otočku zoznamu (po inicializácii, samozrejme, neurobil nič).



Vráti príznak, či má LinkedList ešte ďalší prvok (teda Aktuálny prvok ukazuje na ďalší prvok a iterátor neprešiel celý zoznam). O správnu hodnotu atribúty Prešiel cyklus sa stará metóda Ďalší, ktorú JAVA vyvolá automaticky po volaní metódy MáĎalší. Pred prvým volaním je hodnota tohto atribútu správne inicializovaná konštruktorom iterátora.



Vráti prvok a posunie sa v štruktúre. Platnosť netreba kontrolovať, keďže tá bola skontrolovaná metódou MáĎalší. O jej korektné volanie sa stará JAVA sama.

