

# Динамична реализация на линеен списък

## 1. Определение.

Динамична линейна структура от еднотипни данни. При нея всеки един елемент (без първият и последният) е свързан с два други. Този, който се намира пред разглеждания се нарича „предшественик“ (predecessor), а следващия – „наследник“ (successor). Крайните елементи са свързани с по един елемент. Сами по себе си, елементите представляват структури състоящи се от две или три полета. Броят на полетата зависи от вида на линейния списък. Едното от тях винаги е за данни. Останалото или останалите са за връзка със съседните елементи. Връзките се реализират чрез указатели. Всеки линеен списък задължително има поне една входна точка, от която започва обхождането му.

## 2. Видове:

- *Едносвързани*: с по една връзка между елементите. При тях може да се върви само в една посока: от началото към края.
- *Двусвързани*: с по две връзки между отделните елементи. По тях може да се върви и в двете посоки.
- *Циклични*: При тях началният елемент е и краен. Могат да бъдат едносвързани или двусвързани.

## 3. Предимства и недостатъци спрямо масивите:

- **Линейните списъци използват по-ефективно паметта на компютъра.** Масивът заема област от паметта, която е с константен размер. Няма как да се добавят нови елементи извън неговите предварително зададени граници. Обратно, декларирането на прекалено голям масив води до заемане на излишна памет. При линейните списъци няма такова ограничение. По всяко време могат да се добавят или премахват елементи като заеманата памет съответства на текущия брой на елементите.
- **Добавянето / премахването на елемент от линеен списък е по-просто.** Добавянето / премахването на елементи в масив изисква преместване с една позиция, съответно назад или напред, на всички елементи след избора. При линейните списъци, след достигане на необходимата позиция, е необходимо само да се пренасочи указателя на предшественика.
- **Достъпът до елемент на масив е много пъти по-бърз.** За да се достигне до желан елемент от линеен списък трябва да се премине през всички преди него, понеже достъпът е последователен. При масивите обръщението към всеки един елемент е директно – чрез неговия индекс.

## 4. Графично представяне.

