**37.Заделяне на обекти от динамичната памет. Проблеми породени от взаимодействията между обекти.**Създадените инстанции от даден клас съществуват докато не излязат извън обсега на

действие (scope, края на блока). За да запазиме инстанцията извън scope-а, може да я

създадем, като я заделим от динамичната памет където тя ще стои активна докато не я

освободим ръчно. Заделянето става чрез new, а изтриването чрез delete.

double \*buff = new double[1024];

Cat \*pCat = new Cat(…);

delete [] buff; delete pCat;

Инстанциите биват заделени в паметта, но за да ги достъпим трябва да използваме

указател сочещ към тях. Предимството е, че указателя ще изчезне когато излезе от

scope-а, но не и инстанцията. Достъпът до полета и методи до дадена инстанция чрез

указател става чрез стрелка ‘->’: pCat->speak(); Указателите ни позволяват да ги

насочваме към инстанции от клас който е наследен спрямо типа на указателя. Така

програмата става гъвкава и съвместима. Пример:

Animal \*ptr = new Cat(…);

ptr = new Dog(…);

ptr->speak();

В този случай ще се извика speak() на класът Animal, освен ако той не е виртуален. Ако

е, ще се извика методът на класът Dog наследяващ класа Animal, ако го има. Възможно

е и конвертиране на класа от един тип към друг чрез cast-ване, но е опасно.

При работата със указатели, трябва да се внимава да не загубиме указател сочещ към

дадена инстанция, без да сме освободили инстанцията предварително, иначе се

получава memory leak и те никога в програмата няма да могат да бъдат освободени.