**57.Референтни типове**.

Референтни типове (типове по референция) са указателите, класовете, интерфейсите, делегатите, масивите и опакованите стойностни типове. Физически референтните типове представляват указател към стойност в динамичната памет, но за CLR те не са обикновени указатели, а специ-ални типово-обезопасени указатели. Това означава, че CLR не допуска на един референтен тип да се присвои стойност от друг референтен тип, който не е съвместим с него (т.е. не е същия тип или негов наследник). В резултат на това в .NET езиците грешките от неправилна работа с типове са силно намалени.

**Референтните типове и паметта**

Всички референтни типове се съхраняват в **динамичната памет** (т. нар. **managed heap**), която се контролира от системата за почистване на паметта (garbage collector). Динамичната памет е специално място от паметта, заделено от CLR за съхранение на данни, които се създават динамично по време на изпълнението на програмата. Такива данни са инстанциите на всички референтни типове.

Когато инстанция на референтен тип престане да бъде необходима на програмата, тя се унищожава от системата за почистване на паметта (т. нар. garbage collector).

Когато инстанцираме референтен тип с оператора **new**, CLR заделя място в динамичната памет, където ще стоят данните и един указател в стека, който съдържа адреса на заделеното място. Веднага след това заделената памет се занулява (освен ако програмистът не инициализира заделената променлива, например чрез извикване на подходящ конструктор).

Ако референтен тип (например клас) съдържа член-данни от стойностен тип, те се съхраняват в динамичната памет. Ако референтен тип съдържа член-данни от референтен тип, в динамичната памет се заделят указатели (референции) за тях, а техните стойности (ако не са **null**) също се заде-лят също в динамичната памет, но като отделни обекти.

**Референтните типове и производителността**

Понякога се приема, че заделянето на динамична памет е бърза операция, защото в текущата реализация (.NET Framework 1.1) физически се импле-ментира чрез преместване на един указател. Освобождаването на памет, обаче, е сложна и времеотнемаща операция, която се извършва от време на време от системата за почистване на паметта (garbage collector).

Ако изчислим средното време, необходимо за заделяне и освобождаване на динамична памет, се оказва, че заделянето и освобождаване на стой-ностните типове е значително по-бързо от референтните типове. Когато производителността е важна за нашата система, трябва да се съобразя-ваме с особеностите на стойностните и референтните типове и начина, по който те заделят и освобождават памет.

Глобално погледнато, нещата около управлението на динамичната памет в .NET Framework са доста комплексни, но в тази тема няма да се спираме на тях. По-нататък, в темата за управление на паметта и ресурсите, ще им обърнем специално внимание.