**56.Пакетирани типове (boxed types). Проблеми с достъпа**.

Sтойностните типове се съхраняват в стека на прило-жението и не могат да приемат стойност **null**, докато референтните типове съдържат указател (референция) към стойност в динамичната памет и могат да бъдат **null**.

Понякога се налага на референтен тип да се присвои обект от стойностен тип. Например може да се наложи в **System.Object** инстанция да се запише **System.Int32** стойност. CLR позволява това благодарение на т. нар. "**опаковане**" на стойностните типове (**boxing**).

В .NET Framework стойностните типове могат да се използват без преобразуване навсякъде, където се изискват референтни типове. При нужда CLR опакова и разопакова стойностните типове автоматично. Това спестява дефинирането на обвиващи (wrapper) класове за примитивните типове, структурите и изброените типове, но разбира се, може да доведе и до някои проблеми, които ще дискутираме по-късно.

**Опаковане (boxing) на стойностни типове**

|  |
| --- |
| Опаковането (boxing) е действие, което преобразува стойностен тип в рефере  нция към опакована стойност. То се извършва, когато е необхо-димо да се преобразува стойностен тип към референтен тип, например при преобразуване на **Int32** към **Object**: int i = 5;  object obj = i; // i се опакова  **Особености при опаковането и разопаковането**  При използване на автоматично опаковане и разопаковане на стойности трябва да се имат предвид някои особености:  - Опаковането и разопаковането намаляват производителността. За оптимална производителност трябва да се намали броят на опако- ваните и разопакованите обекти.  - Опакованите типове са копия на оригиналните стойности, поради което, ако променяме оригиналния неопакован тип, опакованото копие не се променя.  При работа с опаковани обекти трябва да се внимава, защото ако не бъдат съобразени някои особености, може да се наблюдава странно пове-дение на програмата . Основната причина за този резултат е фактът, че при преобразуване към интерфейс структурите се опаковат и съответно се създава копие на данните, намиращи се в тях. Опаковането е съвсем в реда на нещата, като се има предвид, че структурите са стойностни типове, а интерфейсите са референтни типове. |