

به نام خدا

محمدجواد قمری

دانشجوی رشته فناوری اطلاعات

درس برنامه نویسی سمت سرور

استاد آقای میثاق پاریان

شماره دانشجویی 01221033720028

تمرین 2

تفاوت int string float bool و ... در حافظه

استفاده از نوع داده صحیح برای متغیر مربوطه مهم است. برای جلوگیری از خطأ، برای صرفه جویی در زمان و حافظه، اما همچنین کد شما را قابل نگهداری و خوانانه تر می کند. رایج ترین انواع داده ها عبارتند از:

Data Type	Size	Description
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
bool	1 bit	Stores true or false values

char	2 bytes	Stores a single character/letter, surrounded by single quotes
string	2 bytes per character	Stores a sequence of characters, surrounded by double quotes

انواع حافظه در رم Stack و Heap و ...

حافظه stack

در بخش **user-space** حافظه قرار دارد و به صورت خودکار توسط CPU مدیریت می شود. متغیر های غیر استاتیک، پارامتر های ارسالی به توابع و آدرس های مربوط به `return` توابع در این حافظه ذخیره می شوند. اندازه حافظه **stack** ثابت است به همین دلیل به آن **static memory** گفته می شود.

در این حافظه اطلاعات پشت سر هم و به ترتیب قرار می گیرند به این صورت که آخرین داده ذخیره شده در بالای **stack** قرار می گیرد و به اصطلاح **push** می شود، حال اگر قصد برداشتن اطلاعات یا به اصطلاح **pop** کردن اطلاعات را داشته باشیم آخرین اطلاعات وارد شده در **stack** را در اختیار داریم. به این الگوریتم **LIFO**(Last In First Out) می گویند. مثال پر کاربرد در توضیح **stack** خشاب اسلحه (آخرین گلوله ای که در خشاب قرار داده می شود اولین گلوله ای است که شلیک می شود) و یا بشقاب های روی هم چیده شده (آخرین بشقابی که روی سایر بشقاب ها قرار داده می شود اولین بشقابی است که برداشته می شود) است.

از آنجا که در حافظه **stack** نیازی به پیدا کردن فضای خالی در حافظه نیست و محل قرارگیری اطلاعات مشخص است (بالای حافظه) بنابراین این حافظه سریع تر از حافظه **heap** است.

حافظه heap

حافظه **Heap** در قسť **user-space** حافظه مجازی قرار دارد و به صورت دستی توسط برنامه نویس مدیریت می شود **Heap**. مربوط به زمان اجرا (**runtime**) است و فضای اشغال شده در **heap** با اتمام کار تابع آزاد نمی شوند و تا زمانی که **Garbage Collector** این فضا را آزاد کند یا توسط برنامه نویس داده ها از حافظه **heap** پاک نشوند در این فضا باقی می ماند. اندازه حافظه **heap** متغیر است به همین دلیل به آن **dynamic memory** گفته می شود.

در این نوع از حافظه برای ذخیره مقادیر ابتدا محاسبه ای توسط سیستم عامل صورت می گیرد تا اولین فضای حافظه ای که اندازه آن متناسب با اندازه ای که مورد نیاز ماست را پیدا کند، در صورت وجود این میزان از حافظه درخواستی آن را به صورت رزرو شده درمی آورد تا بقیه برنامه ها به این فضا

دسترسی نداشته باشند، سپس آدرس ابتدای این فضای محاسبه شده به صورت یک اشاره گر (pointer) در اختیارمان قرار می دهد یا به اصلاح.(allocating)

متغیر ها به صورت پیش فرض در این حافظه قرار نمی گیرند و اگر قصد ذخیره متغیر ها در این حافظه را داشته باشیم باید به صورت دستی این اقدام انجام شود. متغیر هایی که در heap ذخیره می شوند به طور خودکار حذف نمی شوند و باید توسط برنامه نویس و به صورت دستی حذف شوند. به طور کلی مدیریت حافظه heap به صورت دستی توسط برنامه نویس انجام می شود. آرایه های داینامیک در heap ذخیره می شوند.

به دلیل محاسبات برای یافتن آدرس شروع حافظه و در اختیار گرفتن pointer حافظه heap نسبت به stack کنتر است. همچنین اگر داده ها به صورت پشت سر هم در block های حافظه قرار نگرفته باشند (این احتمال بسیار زیاد است) موجب کندی در بازیابی اصلاحات خواهد شد.

کدام متغیر ها reference type هستند؟ کدام متغیر ها value type هستند؟

Value Type

به داده نوعی Value Type گفته میشود که یک مقدار را در فضای حافظه خود ذخیره کند. و این به این معناست که متغیر هایی که از نوع این داده نوع تعریف میشوند به طور مستقیم دارای مقداری در خود هستند.

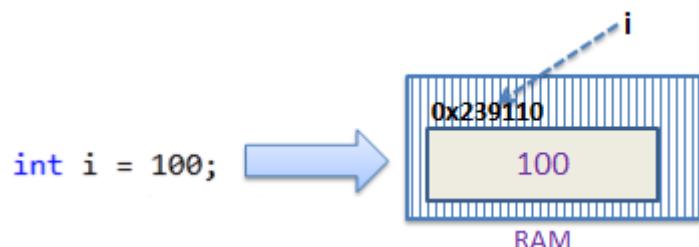
نکته : تمام Value Type ها از فضای نام System.ValueType مشتق میشوند که آن فضای نامی هم در فضای نام System.Object قرار دارد.

برای مثال متغیری از نوع int را در نظر بگیرید :

```
int i = 100;
```

سیستم مقدار عدد صحیح 100 را در فضای حافظه ای که برای متغیر " " اختصاص داده شده است ، ذخیره می کند.

" تصویر زیر نحوه ذخیره سازی مقدار 100 را در حافظه به آدرس (0x239110) برای متغیر " i " نشان میدهد :



همه‌ی داده نوع هایی که در زیر آورده شده است از نوع value type هستند :

int	float	double	enum	decimal	char	byte	bool
ulong		unit		sbyte	sbyte	short	long
							ushort

Reference Type

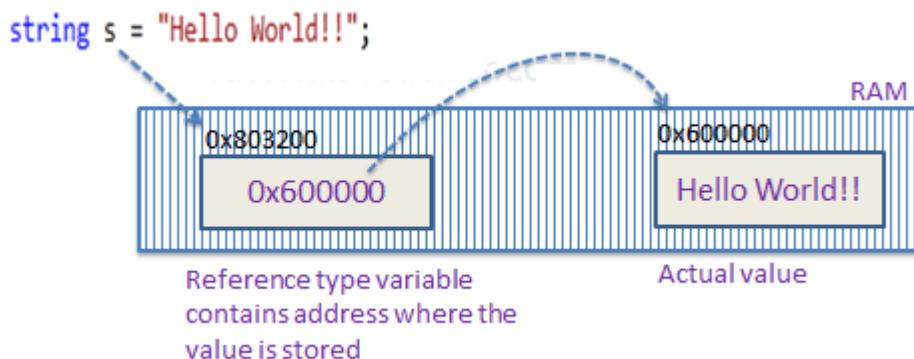
برخلاف Value Type ها، Reference Type ها مقدیرشان را به صورت مستقیم در خود ذخیره نمی‌کنند. در عوض آنها آدرس مکانی از حافظه را که مقدار در آن قرار گرفته است، در خود ذخیره می‌کنند.

به عبارت دیگر Reference Type ها شامل یک اشاره گر هستند که به مکانی دیگر از حافظه اشاره می‌کند که داده یا مقدار در آن ذخیره شده است.

برای این حالت یک متغیر رشته‌ای را می‌توان مثال برد :

```
string s = "Hello Worlds";
```

تصویر زیر چگونگی تخصیص حافظه را برای متغیر رشته‌ای بالا نشان میدهد :



همانطور که در تصویر بالا مشاهده می‌کنید سیستم یک مکان تصادفی در حافظه (0x803200) را برای متغیر `s` انتخاب کرده است. مقداری که در متغیر `s` قرار می‌گیرد `0x600000` است که آدرس خانه‌ای از حافظه است که مقدار اصلی یعنی `Hello Worlds!!` در آن قرار گرفته است.

داده نوع های زیر همگی Reference Type هستند :

تمام آرایه، حتی اگر مقدیر آنها از نوع value type باشد

String

Class

Delegates