

تفاوت `int` — `string` — `bool` — `float` در حافظه چیست؟؟ در برنامه نویسی C باید نوع داده برای هر متغیر تعریف شود. بدین ترتیب نوع و اندازه داده تخصیص داده شده به متغیر هنگام تعریف آن مشخص خواهد شد.

`int`

اعداد صحیح اعداد کامل شامل مقادیر صفر، مثبت و منفی می باشند اما مقدار اعشاری ندارند. به عنوان مثال ۰، ۵، ۱۰،

`double` و `float`

از `float` و `double` برای ذخیره اعداد حقیقی استفاده می شود.

تفاوت `float` و `double` چیست؟

اندازه `float` اعشار با دقت `single` یا واحد (۴ بایت است و اندازه) `double` اعشار با دقت `double` یا مضاعف (۸ بایت است).

`char`

کلمه کلیدی `char` برای تعریف متغیرهای کاراکتری استفاده می شود.

`void`

`void` یک نوع داده ناتمام (`incomplete`) و به معنای "هیچ" یا "بدون نوع" است. می توانید آن را غیرموجود در نظر بگیرید.

به عنوان مثال تابعی که هیچ مقداری برنمی گرداند، نوع بازگشتی آن باید `void` باشد.

توجه داشته باشید نمی توانید متغیری با نوع `void` تعریف کنید.

`long` و `short`

اگر نیاز به استفاده از اعداد بزرگ داشته باشید می توانید از نوع `long` استفاده کنید.

۲) انواع حافظه در رم (stack و heap)؟؟

حافظه stack

در بخش user-space حافظه قرار دارد و به صورت خودکار توسط CPU مدیریت می شود. متغیرهای غیراستاتیک، پارامترهای 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 پرکاربرد 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.

حافظه Heap

حافظه Heap در قسمت 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 در می 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 اصطلاح allocating

۳) کدام متغیرها reference type هستند؟

تمامی نوع های داده درسی شارپ به دو دسته تقسیم می شوند:

Value Types

Reference Types

تفاوت این دو نوع داده در نحوه تخصیص حافظه و مدیریت آن ها می باشد. در بستر .NET زمانی که شما متغیری را تعریف و مقداردهی می کنید بسته به نوع داده این متغیرها می توانند در دو حافظه زیر ذخیره شوند:

Memory Stack

Memory Heap

این دو حافظه یعنی Stack Memory و Heap Memory از نظر میزان سرعت دسترسی و مدیریت آن ها توسط زبان سی شارپ با یکدیگر تفاوت دارند که در ادامه با این حافظه ها بیشتر آشنا می شویم.

متغیرهایی که از نوع Value Type هستند، مستقیماً درون حافظه stack ذخیره می شوند.

نحوه ذخیره سازی متغیرهایی از نوع Reference Type با نحوه ذخیره سازی متغیرهایی از نوع Value Type تفاوت دارد. برخلاف Value

Type ها، Reference Type ها مقادیرشان را به صورت مستقیم در خود ذخیره نمی کنند Reference Type. ها علاوه بر حافظه stack با یک حافظه دیگر به نام Heap نیز سرو کار دارند. حافظه Heap حجم بیشتری نسبت به حافظه Stack در اختیار شما قرار می دهد و از طرفی سرعت دسترسی به آن از حافظه stack کمتر است.