

حافظه های هیپ و استک در برنامه نویسی دو نوع حافظه هستند که تفاوت هایی با یکدیگر دارند. حافظه استک برای ذخیره متغیرهای محلی غیر استاتیک، پارامترهای توابع و آدرس های بازگشتی (return) توابع استفاده می شود و به صورت خودکار توسط CPU مدیریت می شود. متغیرهایی که در stack قرار می گیرند، به طور طبیعی به صورت محلی هستند و زمانی که تابع خارج می شود، تمامی متغیرهای موجود در آن همراه با آن خارج و به پایان زندگی خود می رسند. حافظه هیپ برای ذخیره داده هایی که به صورت داینامیک ایجاد می شوند، مانند آرایه های داینامیک، استفاده می شود و به صورت دستی توسط برنامه نویس مدیریت می شود. متغیرهایی که در heap ذخیره می شوند به طور خودکار حذف نمی شوند و باید توسط برنامه نویس و به صورت دستی حذف شوند. تفاوت های حافظه های هیپ و استک در برنامه نویسی به شرح زیر است:

حافظه Heap :

- در قسمت user-space حافظه مجازی قرار دارد و به صورت دستی توسط برنامه نویس مدیریت می شود.
- مربوط به زمان اجرا (runtime) است و فضای اشغال شده در heap ، توسط برنامه نویس آزاد شده و حذف نمی شود.
- متغیرهایی که در heap ذخیره می شوند به طور خودکار حذف نمی شوند و باید توسط برنامه نویس و به صورت دستی حذف شوند.
- آرایه های داینامیک در heap ذخیره می شوند.
- محدودیتی در اندازه متغیرها ندارد.

حافظه Stack :

- در قسمت user-space حافظه قرار دارد و بصورت خودکار توسط CPU مدیریت می شود.
- محل نگهداری متغیرهای محلی غیر استاتیک، پارامترهای توابع و آدرس های بازگشتی (return) توابع است که بصورت LIFO (Last In First Out) داخل این حافظه بر روی هم انباشته می شوند و به Frame معروفند.
- متغیرهایی که در stack قرار می گیرند، به طور طبیعی به صورت محلی هستند و زمانی که تابع خارج می شود، تمامی متغیرهای موجود در آن همراه با آن خارج و به پایان زندگی خود می رسند.
- هر برنامه دارای یک thread main و هر thread دارای یک حافظه Stack خصوصی است.
- داده ها در حافظه Stack بترتیب بر روی هم قرار می گیرند (با قاعده LIFO)
- با افزایش مصرف حافظه Stack ، حافظه کمتری برای heap باقی می ماند.
- این نوع از حافظه readable و writable است.
- مشکل Stackoverflow نیز در این حافظه به وجود می آید.

متغیرهای Reference Type در برنامه‌نویسی، به متغیرهایی گفته می‌شود که به یک شیء در حافظه اشاره می‌کنند

به عبارت دیگر، مقدار این متغیرها، آدرس حافظه‌ای است که شیء مورد نظر در آن قرار دارد. برای مثال، متغیرهایی که از نوع رشته (string)، آرایه (array)، لیست (list) و شیء (object) هستند، به عنوان متغیرهای Reference Type شناخته می‌شوند. متغیرهای Reference Type در برابر متغیرهای Value Type، دارای رفتار متفاوتی هستند. در متغیرهای Value Type، مقدار متغیر به صورت مستقیم در حافظه ذخیره می‌شود و هر تغییری در مقدار متغیر، مستقیماً بر روی مقدار ذخیره شده در حافظه تأثیر می‌گذارد. اما در متغیرهای Reference Type، مقدار متغیر، آدرس حافظه‌ای است که شیء مورد نظر در آن قرار دارد و هر تغییری در مقدار متغیر، تنها بر روی آدرس حافظه‌ای تأثیر می‌گذارد و مقدار شیء در حافظه تغییر نمی‌کند. به طور کلی، متغیرهای Reference Type در برنامه‌نویسی، به متغیرهایی گفته می‌شود که به یک شیء در حافظه اشاره می‌کنند و مقدار آن‌ها، آدرس حافظه‌ای است که شیء مورد نظر در آن قرار دارد.

نمونه‌هایی از متغیرهای refrence type عبارتند از:

- رشته (string)
- آرایه (array)
- لیست (list)
- شیء (object)

نمونه‌هایی از متغیرهای Value Type عبارتند از:

- اعداد صحیح (integers) مانند int و long
- اعداد اعشاری (floating point numbers) مانند float و double
- مقادیر بولین (boolean values) مانند bool
- مقادیر کاراکتری (character values) مانند char
- ساختارهای داده‌ای (structs) که توسط برنامه‌نویس تعریف می‌شوند.

مزیت اصلی متغیرهای Value Type، سرعت بالای آن‌ها در اجرای برنامه‌هاست. به دلیل اینکه مقدار متغیرهای Value Type به صورت مستقیم در حافظه ذخیره می‌شود، دسترسی به آن‌ها بسیار سریع است و نیازی به مدیریت حافظه توسط برنامه‌نویس نیست. همچنین، متغیرهای Value Type به صورت پیش فرض دارای مقدار اولیه هستند و نیازی به مقداردهی اولیه آن‌ها توسط برنامه‌نویس نیست. در کل، متغیرهای Value Type در برنامه‌نویسی، به متغیرهایی گفته می‌شود که مقدار آن‌ها به صورت مستقیم در حافظه ذخیره می‌شود و هر تغییری در مقدار متغیر، مستقیماً بر روی مقدار ذخیره شده در حافظه تأثیر می‌گذارد.