به نام خدا

تمرین درس برنامه نویسی شی گرایی

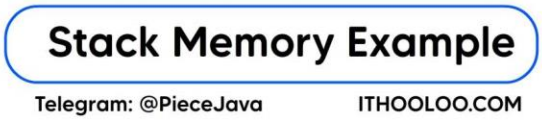
نام استاد : مهندس یاریان

نام دانشجو : محمد مهدی عبداللهی پیربداغ

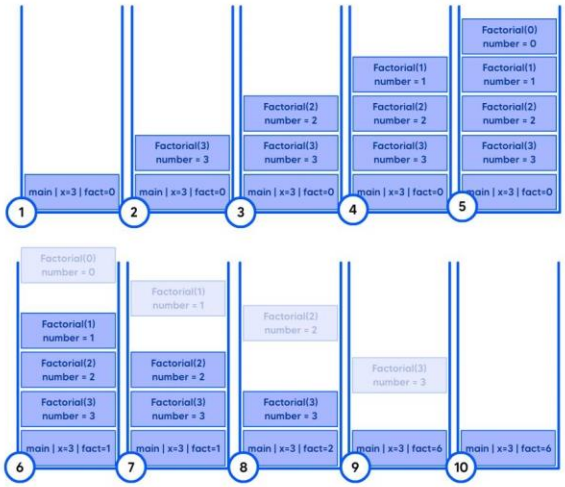
عنوان : حافظه Heap و Stack

حافظه **Heap** و **Stack** بترتیب برای ذخیره اشیا در سطح حافظه و مدیریت داده های محلی و ترتیب اجرای متد ها در جاوا کاربرد دارند.

حافظه استک در واقع فضاییست که به اجرا شدن توابع، داده های نوع پیشین، متغیر های محلی و متغیر های مرجع موجود در توابع اختصاص داده شده است. دسترسی به حافظه استک بصورت خروج به ترتیب عکس ورود یا Last in First Out میباشد. هر متدی که اجرا میشود در حافظه استک یک بلاک حافظه به آن اختصاص داده میشود و هر داده پیشین و ارجاع به آبجکت های موجود در بدنه متد، در این بلاک حافظه ذخیره میشود. وقتی متد بصورت کامل اجرا میشود، بلاک حافظه از حافظه استک حذف و این حافظه برای استفاده مجدد در دسترس قرار میگیرد.

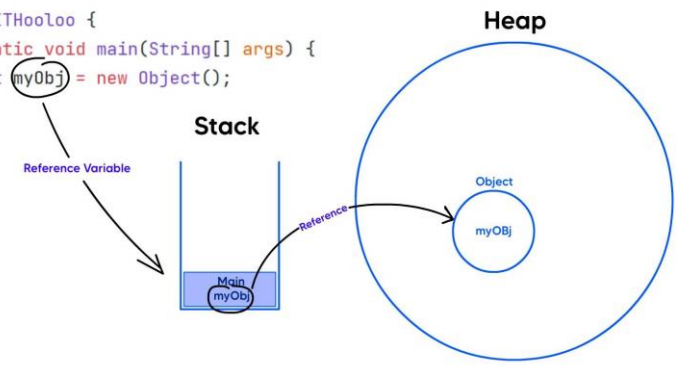



```java
public class ITHooloo {
    public static void main(String[] args) {
        int x = 3;
        int fact = factorial(x);
    }
    static int factorial(int number){
        if(number==0)
            return 1;
        return number*factorial( number: number-1);
    }
}
```



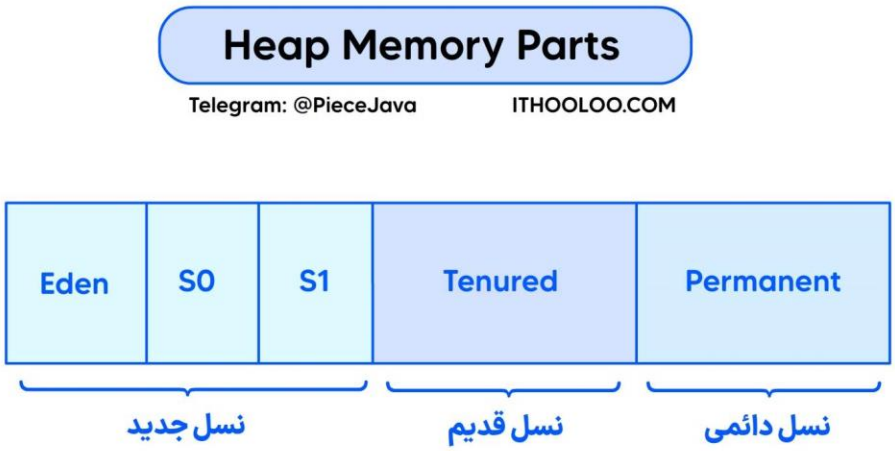
حافظه **هیپ** در ذخیره آبجکت های ایجاد شده در زمان اجرای یک برنامه **جاوا** بسیار موثر است. البته مسیر ارجاع به آبجکت ایجاد شده همچنان در حافظه استک ذخیره میشود که به همان آبجکت ایجاد شده در حافظه هیپ اشاره میکند. حافظه هیپ در مقایسه با حافظه استک، فضای خیلی بزرگتری در اختیار دارد. در حافظه هیپ ابجکت های بلااستفاده توسط **گاربیج کالکتور** برای ازاد سازی حافظه حذف میشوند.


```java
public class ITHooloo {
    public static void main(String[] args) {
        Object myObj = new Object();
    }
}
```



تقسیم بندی حافظه هیپ

حافظه هیپ به سه بخش تقسیم میشود: نسل جدید، نسل قدیم و نسل دائمی.



نسل جدید

نسل جدید بخشی است که به آبجکت های تازه ایجاد شده اختصاص داده میشود. نسل جدید نیز به سه بخش **Eden** و **Survivor ۱** و Survivor ۲ تقسیم میشود. در ابتدا تمامی آبجکت ها در **داخل حافظه Eden**قرار گرفته میشوند. پس از پر شدن حافظهEden ، یک گاربیج کالکشن جزئی اتفاق می افتد تا آبجکت های بلااستفاده از سطح حافظه حذف شوند. در این حالت آبجکت های باقیمانده به بخش Survivor ۱ و سپس به بخش Survivor ۲ انتقال میابند.

نسل قدیم

در ادامه آبجکت های باقیمانده به بخش نسل قدیم انتقال می یابند. در این بخش معمولا کمی از ابجکت ها دوباره توسط گاربیج کالکشن جمع اوری میشوند تا این اطمینان حاصل شود که در این بخش فقط ابجکت های ماندگار وجود دارند.

نسل دائمی

در این بخش JVM برای ذخیره متادیتا (فراداده) در مورد کلاس ها و متد ها استفاده میکند. البته یک نظر اینجا وجود دارد که بیان میکند نسل دائمی جزئی از حافظه هیپ نیست و به یک بخش دیگر اختصاص دارد.