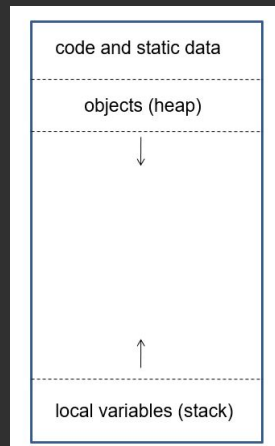


Uvod u objektno orijentirano programiranje

Memorija

Struktura memorije

- Osnovna jedinica memorije je **bit**, 0 ili 1
- Kod većine arhitektura najmanja jedinica na kojoj hardware može vršiti operacije je **bajt**, niz od osam bitova
- U JVM, svaka deklaracija varijable, objekta, poziv new metode i slično se sprema u "stack memory" ili u "heap space"



Stack Memory

- Koristi se za statičku alokaciju i izvođenje dretva
- Sadrži primitivne vrijednosti koje su specifične za metode i reference objekata koji se nalaze u heap-u
- LIFO pristup
- Povećava se i smanjuje s pozivom metode i sa završetkom metode
- Varijable unutra postoje onoliko dugo koliko se izvršava metoda u kojoj su stvorene
- Automatska alokacija i delokacija
- Ako je memorija popunjena dobivamo `java.lang.StackOverFlowError`
- Brzi pristup s obzirom na heap
- Threadsafe

Heap Space

- Koristi se za dinamičku alokaciju tijekom izvođenja
- Novi objekti se uvijek stvaraju u heap-u, a reference na objekt ide u stack
- Dijelovi heap-a :
 - **Young Generation** - Svi novi objekti, Garbage collection kada se napuni, (podjeljena u tri dijela)
 - **Old Generation** - Kada se dosegne threshold "življenja" u Young Generation, objekt se seli u Old Generation
 - **Permanent Generation** - JVM metapodatci
- Ako je memorija popunjena dobivamo `java.lang.OutOfMemoryError`
- Nema automatske delokacije, Garbage Collector čisti memoriju
- Nije threadsafe i treba paziti na sinkronizaciju koda

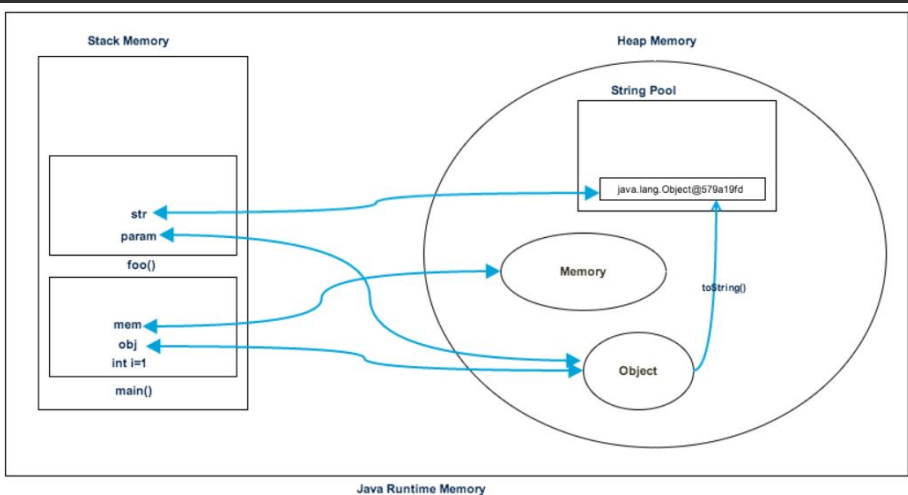
Ostali dijelovi memorijskog modela

- Method Area - konstante i statičke varijable (Runtime Constant Pool), kod za metode i konstruktore
- Memory Pool - "immutable" objekti (String pool)
- GC

Java Garbage Collection

- Proces identifikacije i uklanjanja nepotrebnih objekata iz memorije
- Za razliku od ostalih programskih jezika, Java nudi automatski GC
- GC je program koji se vrti u pozadini i traži objekte koji nisu referencirani
- Možemo definirati i tip GC-a

```
public class Memory {  
  
    public static void main( String[] args) {  
        int i=1;  
        Object obj = new Object();  
        Memory mem = new Memory();  
        mem.foo(obj);  
    }  
  
    private void foo( Object param) {  
        String str = param.toString();  
    }  
  
}
```



Zadatak za vježbu

- Napišite metodu `momakDjevojka` koja kao parametar prima čitač određene datoteke npr. `(BufferedReader r = new BufferedReader(new FileReader("in.txt"));``momakDjevojka(r))`. U datoteci je zapis slijedećeg oblika: "Maja 34 Roko 3 Nina 2 Slaven 42 ". Potrebno je zbrojiti brojeve uz muška i ženska imena (izmjenjuju se, prvo je žensko ime), npr: Momci : 45 Djevojke : 36. Datoteka može završiti s imenom djevojke ili momka.