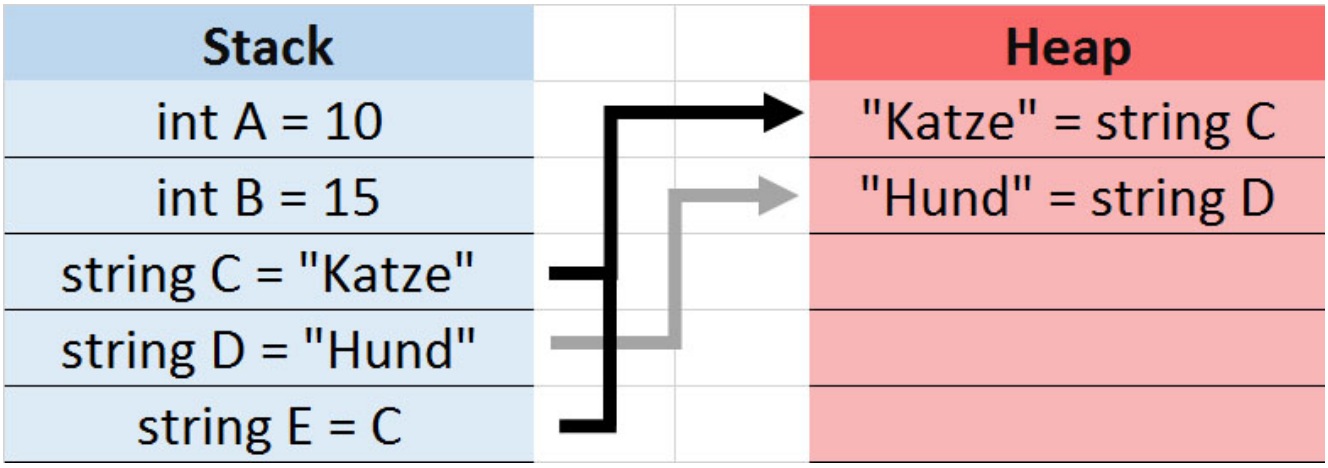
Referenztypen, Wertetypen

Stack und Heap sind Teil des Arbeitsspeichers, die zu mindestens unter C# vom Kompiler automatisch verwaltet wird.

Beim **Stack** (dt. Stapel) werden die Daten quasi aufeinander gelegt. Wenn Speicher freigegeben werden kann, so wird dies auch von oben heraus getan (LIFO Prinzip). Durch dieses Prinzip ist der Stack sehr schnell in seiner Arbeitsweise. Verlässt der Programmcode die geschwungene Klammer { }, werden sämtliche innerhalb angelegte Variablen aus dem Stack entfernt. Obwohl der Speicher sehr gering ist, wächst und schrumpft er während des Programmablaufs.

Der **Heap** ist nicht so strukturiert wie der Stack, dort liegen die Daten quasi durcheinander und werden vom Stack aus per Zeiger gefunden. Deswegen ist der Heap auch deutlich langsamer in seiner Arbeitsweise.

Was in den Stack und was in den Heap gelangt, bestimmen unter anderem die Datentypen. Weiter beinhaltet der Stack auch die Zeiger der Variablen auf den Heap.



Stack – Wertetypen:

• Alle numerischen Datentypen

• Boolean, Char und Date
• Alle Strukturen, auch wenn ihre Member Verweistypen sind
• Enumerationen, da der zugrunde liegende Typ immer SByte, Short, Integer, Long, Byte, UShort, UInteger oder ULong ist

Heap – Referenztypen, Verweistypen:

• String
• Alle Arrays, auch wenn ihre Elemente Wertetypen sind
• Klassentypen, z.B. Form
• Delegaten
• reine Objekte

Das Verschieben der Daten vom Stack (Wertetyp) zu Heap (Referenztyp) bezeichnet man als **boxing**
[crayon-6606b7110409b490865836/]
und vom Heap zu Stack als **unboxing**
[crayon-6606b711040a3222313713/]

Auch beim **casten** können die Daten vom Heap zum Stack geschoben werden:
[crayon-6606b711040a4305210866/]

Quelle:

Understanding Boxing and Unboxing