KSP Aufgabe 5

1. Die Aufgabenstellung hier ist ganz einfach, bedingt aber einige Änderungen in der VM: Verlagern Sie die Rechenobjekte (bis jetzt sind das ja nur ganze Zahlen) auf den Heap und halten Sie in der Static Data Area, auf dem Stack und im Rückgabewertregister nur noch Zeiger in den Heap. Der Speicherplatz für die Objekte wird mit `malloc()` beschafft und innerhalb Ihres Programms nicht wieder freigegeben (das ist nämlich ohne Garbage Collector praktisch nicht möglich, und wie ein solches Teil funktioniert, werden wir in der Vorlesung erst später behandeln). Was genau sind unsere "Rechenobjekte"? Nun, das ist einfach: alle Objekte in der Static Data Area, alle Objekte auf dem Stack (mit Ausnahme von Rücksprungadressen und Framepointerwerten) sowie der Inhalt des Rückgabewertregisters. Alle anderen in der VM benutzten Dinge, wie z.B. Codespeicher, Programmzähler, Stackpointer und Framepointer sind keine Rechenobjekte: man kann sie von einem Ninja-Programm aus nicht unmittelbar beeinflussen.
2. Der Instruktionssatz ist gegenüber Aufgabe 4 unverändert; auch der Compiler und der Assembler bleiben exakt gleich (bis auf die Versionsnummern).
3. Ihr Debugger muss beim Anzeigen des Stacks kenntlich machen, welche Einträge Objektreferenzen und welche Einträge einfache Zahlen (Rücksprungadressen und gespeicherte Framepointer) sind.
4. Bauen Sie in Ihren Debugger eine Möglichkeit ein, Objekte zu inspizieren. Warum? Wenn Sie z.B. wissen wollen, welche Zahl ganz oben auf dem Stack liegt, dann lassen Sie sich den Stack anzeigen. Dort finden Sie aber nur die Adresse des Objektes auf dem Heap. Also wünschen Sie sich eine Möglichkeit, den Inhalt eines Objektes bei gegebener Objektadresse anzuschauen. Das wird im Übrigen auch in der übernächsten Aufgabe gebraucht, wenn in Objekten Referenzen auf andere Objekte gespeichert werden.
5. Hier wie immer die Referenzimplementierung: njvm