```
class Memory {
  static Array memory;
  static int free;

  /** Initializes memory parameters. */
  function void init() {
    let memory = 0;
    let free = 2048;
    return;
  }

  /** Returns the value of the main memory at the given address. */
  function int peek(int address) {
    return memory[address];
  }

  /** Sets the value of the main memory at this address
   *  to the given value. */
  function void poke(int address, int value) {
    let memory[address] = value;
    return;
  }

  /** finds and allocates from the heap a memory block of the
   *  specified size and returns a reference to its base address. */
  function int alloc(int size) {
    var int pointer;

    if (size < 1) {
      do Sys.error(5); // Memory.alloc: Allocated memory size must be positive
    }

    let pointer = free;
    let free = free + size;

    if (free > 16383) {
      do Sys.error(6); // Memory.alloc: Heap overflow
    }

    return pointer;
  }

  /** De-allocates the given object and frees its space. */
  function void deAlloc(int object) {
    // add to list of available blocks
    return;
  }
}
```