```
class Screen {
  static boolean color;
  static Array bit_masks;

  /** Initializes the Screen. */
  function void init() {
    let color = true;

    let bit_masks = Array.new(16);
    let bit_masks[0]  = 1;
    let bit_masks[1]  = 2;
    let bit_masks[2]  = 4;
    let bit_masks[3]  = 8;
    let bit_masks[4]  = 16;
    let bit_masks[5]  = 32;
    let bit_masks[6]  = 64;
    let bit_masks[7]  = 128;
    let bit_masks[8]  = 256;
    let bit_masks[9]  = 512;
    let bit_masks[10] = 1024;
    let bit_masks[11] = 2048;
    let bit_masks[12] = 4096;
    let bit_masks[13] = 8192;
    let bit_masks[14] = 16384;
    let bit_masks[15] = 16384 + 16384;

    return;
  }

  /** Erases the whole screen. */
  function void clearScreen() {
    do Screen.setColor(false);
    do Screen.drawRectangle(0, 0, 511, 255);
    do Screen.setColor(true);

    return;
  }

  /** Sets the color to be used in further draw commands
   *  where white = false, black = true. */
  function void setColor(boolean b) {
    let color = b;

    return;
  }

  /** Draws the (x, y) pixel. */
  function void drawPixel(int x, int y) {
    var int row, col, bit, address, value;
```

```
    let col = x;
    let row = y;

    if ((col < 0) | (col > 511) | (row < 0) | (row > 255)) {
      do Sys.error(7); // Screen.drawPixel: Illegal pixel coordinates
    }

    let bit = col - ((col / 16) * 16); // col % 16

    let address = 16384 + (row * 32) + (col / 16);

    let value = Memory.peek(address);

    if (color) {  // black
      let value = value | bit_masks[bit];
    } else {      // white
      let value = value & ~bit_masks[bit];
    }

    do Memory.poke(address, value);

    return;
  }

  /** Draws a line from (x1, y1) to (x2, y2). */
  function void drawLine(int x1, int y1, int x2, int y2) {
    var int a, b, a_inc, b_inc, dx, dy, criterion, temp;

    if ((x1 < 0) | (x1 > 511) | (y1 < 0) | (y1 > 255) |
        (x2 < 0) | (x2 > 511) | (y2 < 0) | (y2 > 255)) {
      do Sys.error(8); // Screen.drawLine: Illegal line coordinates
    }

    let a = 0;
    let b = 0;

    if (~(x1 < x2)) {
      let temp = x1;
      let x1 = x2;
      let x2 = temp;

      let temp = y1;
      let y1 = y2;
      let y2 = temp;
    }

    let dx = x2 - x1;
    let dy = y2 - y1;
```

```
    // do Screen.drawPixel(a, b);

    if (dy < 0) {
      let b_inc = -1;
    } else {
      let b_inc = 1;
    }

    if (dy = 0) {
      while (~(a = dx)) {
        let a = a + 1;
        do Screen.drawPixel(x1 + a, y1);
      }

      return;
    }

    if (dx = 0) {
      while (~(b = dy)) {
        let b = b + b_inc;
        do Screen.drawPixel(x1, y1 + b);
      }

      return;
    }

    while ((a < dx) & (b < dy)) {
      let criterion = (a * dy) - (b * dx);

      if (criterion < 0) {
        let a = a + 1;
      } else {
        let b = b + 1;
      }

      do Screen.drawPixel(x1 + a, y1 + b);
    }

    while ((a < dx) & (b > dy)) {
      let criterion = (a * dy) - (b * dx);

      if (criterion > 0) {
        let a = a + 1;
      } else {
        let b = b - 1;
      }

      do Screen.drawPixel(x1 + a, y1 + b);
    }

```

```
    return;
  }

  /** Draws a filled rectangle where the top left corner
   *  is (x1, y1) and the bottom right corner is (x2, y2). */
  function void drawRectangle(int x1, int y1, int x2, int y2) {
    var int temp, dx, dy, a, b;

    if ((x1 < 0) | (x1 > 511) | (y1 < 0) | (y1 > 255) |
        (x2 < 0) | (x2 > 511) | (y2 < 0) | (y2 > 255) |
        (x1 > x2) | (y1 > y2)) {
      do Sys.error(9); // Screen.drawRectangle: Illegal rectangle coordinates
    }

    let a = 0;
    let b = 0;

    let dx = x2 - x1;
    let dy = y2 - y1;

    while (b < dy) {
      while (a < dx) {
        do Screen.drawPixel(x1 + a, y1 + b);

        let a = a + 1;
      }

      let a = 0;
      let b = b + 1;
    }

    return;
  }

  /** Draws a filled circle of radius r around (cx, cy). */
  function void drawCircle(int cx, int cy, int r) {
    var int dy, sqrt, x1, x2, y;

    if ((cx < 0) | (cx > 511) | (cy < 0) | (cy > 255)) {
      do Sys.error(12); // Screen.drawCircle: Illegal center coordinates
    }

    if ((r < 0) | (r > 127) | ((cx - r) < 0) | ((cx + r) > 511) |
      ((cy - r) < 0) | ((cy + r) > 255)) {
      do Sys.error(13); // Screen.drawCircle: Illegal radius
    }

    let dy = -r;

    while (~(dy = r)) {
```

```
      let sqrt = Math.sqrt((r * r) - (dy * dy));

      let x1 = cx - sqrt;
      let x2 = cx + sqrt;

      let y = cy + dy;

      do Screen.drawLine(x1, y, x2, y);

      let dy = dy + 1;
    }

    return;
  }
}
```