Prof. O. Nierstrasz
Toon Verwaest, Camillo Bruni

Serie 2 - Postscript

Exercise 1

- Create groups of 2 and inform us.
- Clone the git repository at `git@pinocchio.unibe.ch:pl-exercises.git`.
- Clone your group's git repository (sent to you by email).
- Solve the exercises.
- Push the solutions (code and text) in the subdir `S02` back to the server before midnight, the night before the next lecture.

From now on the exercises will only appear in the pl-exercises git repository. In case you have questions please ask them on the mailing list.

Exercise 2

Answer the following questions about Postscript:

- How do you manipulate the coordinate system?
- Why would you define your own dictionaries?
- When should you use `translate` instead of `moveto`?
- When would you use a matrix instead of `gsave`/`grestore`?
- Why is it important to leave the stack in a consistant state?
- Implement the equivalent of the following piece of code in postscript:

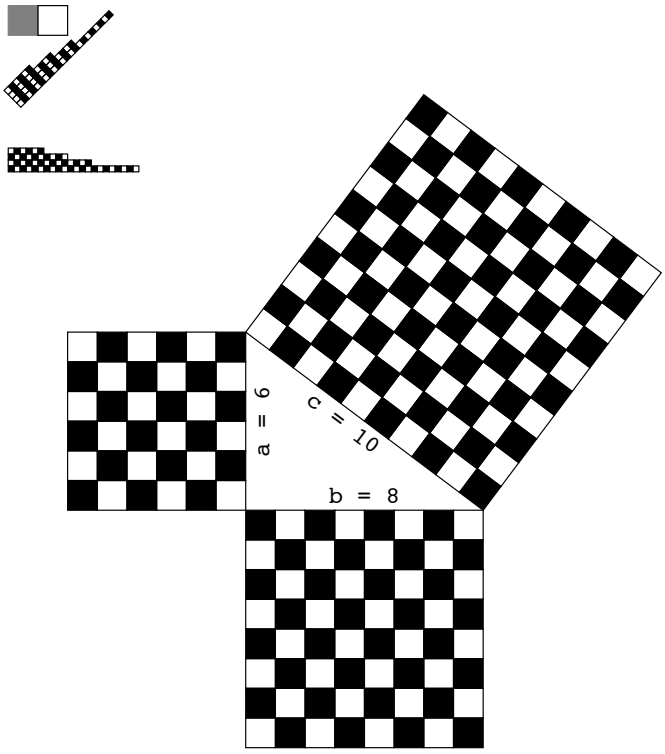
```
public int f(int a, int b) {
    int d = x(a, b);
    z(a, b);
    return d;
}

public int x(int a, int b) {
    return a - b;
}

public int z(int a, int b) {
    return a + b;
}
```

Exercise 3

Write the procedures to get the following drawings:



The exercises consist of the following steps:

1. Create a procedure `/box` drawing a rectangle of a given size
2. Create a procedure `/filledbox` and `/borderedbox` both drawing a rectangle, one completely filled, the other only with outline drawn. Both procedures expect a gray color and a size.
3. Create a procedure `/drawline` drawing a line of alternating filled and bordered rectangles. It expects the size of each rectangle and the number of rectangle pairs as arguments.
4. Create the procedures `/evenline` and `/oddline` both working similar to `/drawline` but one starting with a filled rectangle and the other with a outlined rectangle. Both take the same arguments as `/drawline`
5. Create a procedure `/chessboard` drawing vertically alternating `/evenline` and `/oddline`. The output should be a rectangular chessboard. It takes the number of pairs of even/oddlines and the size of a single rectangle as argument.