

Aufgabe 1: Abbildung von Exceptions

Gegeben sei die folgende Java-Klasse:

```
class TryFinally {
    public static int test1() {
        try {
            return 1;
        } finally {
            return 2;
        }
    }

    public static int test2() {
        int x = 1;
        try {
            return x;
        } finally {
            x = 2;
        }
    }

    public static int test3() {
        while (true) {
            try {
                return 1;
            } finally {
                break;
            }
        }
        return 2;
    }

    public static int test4() {
        for (int i = 10; i < 20; ) {
            try {
                return ++i;
            } finally {
                continue;
            }
        }
        return 2;
    }

    public static void main(String[] args) {
        System.out.println(test1());
        System.out.println(test2());
        System.out.println(test3());
        System.out.println(test4());
    }
}
```

1.1 Ausgabe

Wie lautet die Ausgabe des Programms? Überprüfen Sie ihre Lösung mit Hilfe eines Java-Compilers.

1.2 Abbildung

Erarbeiten Sie anhand der letzten Teilaufgabe die Abbildung von `try` und `finally` auf einen CFG.

Aufgabe 2: Java Bytecode

Eine detaillierte Beschreibung der Java Virtual Machine (und Java Bytecode) finden Sie in der „The Java™ Virtual Machine Specification“.

2.1 Übersetzen

Übersetzen Sie den Ausdruck `this.x = 32 - this.foo()[2]` in Java Bytecode. Geben Sie auch einen passenden Konstantenpool an. `foo` ist eine nicht-statische Methode.

2.2 Wissen

- Was ist der Unterschied zwischen `aload 2` und `aload_2`?
- Mit den Befehlen `bipush` und `sipush` kann man 8- oder 16-Bit-Immediate-Werte auf den Stack legen. Warum gibt es keine Variante für 32-Bit-Zahlen? Was tut man stattdessen?
- Was ist der Unterschied zwischen `ret` und `return`?

2.3 Fehlersuche

Das folgende Java Bytecode Programm soll ABCDEFGHIJKLMNOPQRSTUVWXYZ ausgeben. Leider haben sich einige Fehler eingeschlichen. Beschreiben und beheben Sie diese Fehler.

```
.class public print
.super java/lang/Object

.method public static main([Ljava/lang/String;)V
    .limit stack 5
    bipush 65                ; start with 65 ('A')

loop:
    ; use System.out.append() to print the character
    ; Signature: "PrintStream PrintStream.append(char c)"
    getstatic java/lang/System/out Ljava/io/PrintStream;
    invokevirtual java/io/PrintStream/append(C)Ljava/io/PrintStream;

    iconst_1
    iadd                    ; increment

    bipush 90                ; if greater than 90 ('Z') jump to end
    if_icmpgt end

    goto loop                ; next loop iteration

end:
    ; call System.out.println() to produce a linebreak
    ; Signature: "void PrintStream.println()"
    getstatic java/lang/System/out Ljava/io/PrintStream;
    invokevirtual java/io/PrintStream/println()V
    return
.end method
```