Arbeitsblatt 4, infT

Copy Constructor

```
#include <iostream>
using namespace std;

class Test {
public:
  Test() {
    cout << "  Test() called\n" << endl;
  }
  Test(const Test&) {
    cout << "  Test(const Test&) called\n" << endl;
  }
  ~Test() {
    cout << "  ~Test() called" << endl;
  }
};

const Test& passOn(const Test& test) {
  return test;
}

const Test copyOut(const Test& test) {
  return test;
}

const Test copyInAndOut(const Test test) {
  return test;
}

int main() {
  cout << "Test t;\n";
  Test t;

  cout << "Test& t2 = t;\n";
  Test& t2 = t;

  cout << "Test t3 = t2;\n";
  Test t3 = t2;

  cout << "t2 = t3;\n";
  t2 = t3;

  cout << "passOn(t3);\n";
  passOn(t3);

  cout << "passOn(passOn(t3));\n";
  passOn(passOn(t3));

  cout << "copyOut(t3);\n";
  copyOut(t3);

  cout << "Test t4 = copyOut(t3);\n";
  Test t4 = copyOut(t3);

  cout << "t4 = copyOut(t2);\n";
  t4 = copyOut(t2);

  cout << "t4 = copyInAndOut(t2);\n";
  t4 = copyInAndOut(t2);

  cout << "about to leave main\n";
  return 0;
}
```

Kommentieren Sie den Output zeilenweise:

```
Test t;
  Test() called
Test& t2 = t;
Test t3 = t2;
  Test(const Test&) called
t2 = t3;
passOn(t3);
passOn(passOn(t3));
copyOut(t3);
  Test(const Test&) called
  ~Test() called
Test t4 = copyOut(t3);
  Test(const Test&) called
t4 = copyOut(t2);
  Test(const Test&) called
  ~Test() called
t4 = copyInAndOut(t2);
  Test(const Test&) called
  Test(const Test&) called
  ~Test() called
  ~Test() called
about to leave main
  ~Test() called
  ~Test() called
  ~Test() called
```