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

class zk {
     char *s;
public:
     zk(const char * const z = 0):s(z ? strcpy(new char[strlen(z)+1], z) : 0){}

     // Kopierkonstruktor
     zk(const zk &zkd):s(&zkd && zkd.s ? strcpy(new char[strlen(zkd.s)+1], zkd.s) : 0){}

     // Zuweisungsoperator ueberladen
     zk &operator=(const zk &zkd){
          if(this && &zkd && &zkd != this){
               delete [] s; s=0;
               s=zkd.s ? strcpy(new char[strlen(zkd.s)+1], zkd.s) : 0;
          }
          return *this;
     }

     ~zk(){
          if(this){ /*cout<<"s = "<<(this->s?this->s:"0")<<endl;*/
               delete [] this->s; this->s = 0;
          }
     }

     static zk cat(const zk zk1, const zk zk2){ //(const zk &zk1, const zk &zk2)
          zk z;
          if(!&zk1 && !&zk2) return z;
          if(!&zk1){ zk z2(zk2); return z2; }
          if(!&zk2){ zk z1(zk1); return z1; }
          z.s = new char[(zk1.s?strlen(zk1.s):0) + (zk2.s?strlen(zk2.s):0) +1];
          z.s[0] = '\0';
          if(zk1.s) strcat(z.s, zk1.s);
          if(zk2.s) strcat(z.s, zk2.s);  /*cout<<"strlen(z.s) = "<<strlen(z.s)<<endl;*/
          return z;
     }
```

```
     zk &cat(const zk &zk1){
          if(!this || !&zk1) return *this;
          char *t = new char[(s?strlen(s):0)+(zk1.s?strlen(zk1.s):0)+1];
          t[0]='\0';
          if(s) strcat(t,s);
          if(zk1.s) strcat(t,zk1.s);
          delete [] s; s = t;
          return *this;
     }

     const char *const get_s(){ return this?s:0; }  // s und *s sind const !!

     void set_s(char *z=0){ if(this){ delete [] s; this->s = z?strcpy(new char[strlen(z)+1], z) : 0; }}

     //Ueberladen des += - Operators
     zk &operator+=(const zk &zkd){
          if(!this || !&zkd) return *this;
          if(s && zkd.s){
               char *t = new char[strlen(s)+strlen(zkd.s)+1];
               strcpy(t, s); strcat(t, zkd.s);
               delete [] s; s = t;
          }
          if(!s && zkd.s){ s = zkd.s?strcpy(new char[strlen(zkd.s)+1], zkd.s):0; }
          return *this;
     }
};

void main(){
     zk *p1 = new zk("HTW"), *p2 = new zk(" Dresden");
     zk z(zk::cat(*p1, *p2));
     const char * const s = z.get_s();
     cout<<(s?s:"0")<<endl;
     const char * const t = p1->cat(*p2).get_s();
     cout<<(t?t:"0")<<endl;
     delete p1; p1=0; delete p2; p2=0;
     cin.get();
}
```