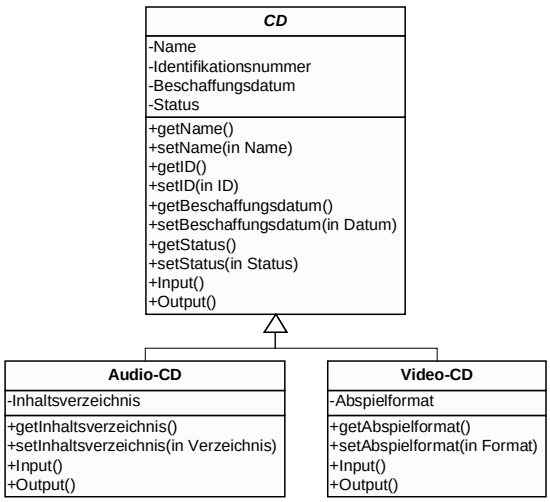
Uwe Grohnwaldt
Yichen Dong
Johannes Rössel

1.



2.

Ucd.pas

```
{$MODE Delphi} // fuer Free Pascal
unit UCd;

interface
type stringarray = array of string;
type TCd = class
    private
      Name    : string;
      IdNr    : longint;
      BDate   : string;
      Status  : String;
    public
      function getName():string;
      procedure setName(Name:string);
      function getIdNr():longint;
      procedure setIdNr(IdNr:longint);
      function getBDate():string;
      procedure setBDate(BDate:string);
      function getStatus():String;
      procedure setStatus(Status:String);
      procedure Input(); virtual;
      procedure Output(); virtual;
  end;

  TAudioCd = class(TCd)
    private
      Toc   : stringarray;
    public
      function getToc() : stringarray;
      procedure setToc(Toc:stringarray);
      procedure Input(); override;
      procedure Output(); override;
  end;

  TVideoCd = class(TCd)
    private
      Format  : string;
    public
      function getFormat() : string;
      procedure setFormat(Format:string);
      procedure Input(); override;
```

```
      procedure Output(); override;
  end;

implementation

function TCd.getName():string;
begin
  getName := self.Name;
end;

procedure TCd.setName(Name:string);
begin
  self.Name := Name;
end;

function TCd.getIdNr():longint;
begin
  getIdNr := self.IdNr;
end;

procedure TCd.setIdNr(IdNr:longint);
begin
  self.IdNr := IdNr;
end;

function TCd.getBDate():string;
begin
  getBDate := self.BDate;
end;

procedure TCd.setBDate(BDate:string);
begin
  self.BDate := BDate;
end;

function TCd.getStatus():String;
begin
  getStatus := self.Status;
end;

procedure TCd.setStatus(Status:String);
begin
  self.Status := Status;
end;

function TAudioCd.getToc() : stringarray;
begin
  getToc := self.Toc;
end;

procedure TAudioCd.setToc(Toc:stringarray);
begin
  self.Toc := Toc;
end;

function TVideoCd.getFormat() : string;
begin
  getFormat := self.Format;
end;

procedure TVideoCd.setFormat(Format:string);
begin
  self.Format := Format;
end;

procedure TCd.Input();
begin
  write('Name: ');
  readln(self.Name);
  write('IdNr: ');
```

```
  readln(self.IdNr);
  write('BDate: ');
  readln(self.BDate);
  write('Status: ');
  readln(self.Status);
end;

procedure TCd.Output();
begin
  writeln('Name: ', self.Name);
  writeln('IdNr: ', self.IdNr);
  writeln('BDate: ', self.BDate);
  writeln('Status: ', self.Status);
end;

procedure TAudioCd.Input();
var
  n : integer;
begin
  inherited Input();
  write('Toc: ');
  write('Bitte die Anzahl der Titel angeben:');
  readln(n);
  setlength(self.Toc, n);
  for n := 0 to Length(self.Toc)-1 do
    begin
      writeln('Bitte Titel Nr. ', n+1,' eingeben');
      readln(self.Toc[n]);
    end;
end;

procedure TAudioCd.Output();
var n:integer;
begin
  inherited Output();
  writeln('Toc:');
  for n:=0 to Length(self.Toc)-1 do
  begin
    writeln(n+1, '. ', self.Toc[n]);
  end;
end;

procedure TVideoCd.Input();
begin
  inherited Input();
  write('Format: ');
  readln(self.Format);
end;

procedure TVideoCd.Output();
begin
  inherited Output();
  write('Format: ');
  readln(self.Format);
end;

end.
```

3.

tree.pas

```
{ Diese Unit basiert auf den Bemuehungen von im Wesentlichen Uwe Grohnwaldt,
  Andreas Daehn und Johannes Roessel. Es ist daher nicht so verwunderlich,
  dass sie in mehreren Aufgabenloesungen vorkommt, in der Hoffnung, dass dies
  in Ordnung ist. Sie fundiert zum Groszteil auf den bei Prof. Kirste
  behandelten Grundlagen fuer Baeume. }
```

```
unit tree;

interface

// brauchen wir wegen unserer Elemente
uses UCd;

type Elem = TCd;
     PElem = ^Rec;
     Rec = record
             val : Elem;
             left, right : PElem;
           end;
     Baum = PElem;
     VProc   = procedure(e : Elem);

function cons(v : Elem; tl, tr : Baum):Baum;
procedure Insert(var t: Baum; e: Elem);
procedure Traverse(t: Baum; p : VProc);
function lookup(t: Baum; name: String): Elem;

implementation

// erstellt einen neuen (Teil-)Baum aus Wert sowie linkem und rechtem Teilbaum
function cons(v : Elem; tl, tr : Baum):Baum;
var t : Baum;
begin
    new(t);
    if t <> nil then begin
        t^.val := v;
        t^.left := tl;
        t^.right := tr;
    end;
    cons := t
end;

// Traversiert den Baum in-order mit der gegebenen Funktion, die fuer jedes
// Element aufgerufen wird
procedure Traverse(t: Baum; p: VProc);
begin
  if t=nil then exit;
  //behandeln des linken teilbaums
  Traverse(t^.left, p);
  p(t^.val);
  Traverse(t^.right, p);
end;

// Einfuegen eines Elements
procedure Insert(var t: Baum; e: Elem);
begin
  if t = nil then t := cons(e, nil, nil)
  else if e.getName() < t^.val.getName() then insert(t^.left, e)
  else if e.getName() > t^.val.getName() then insert(t^.right, e)
  else writeln('Element schon vorhanden');
end;

// Suche nach einem Element
function lookup(t: Baum; name: String): Elem;
begin
  if t = nil then lookup := nil
  else if t^.val.getName() = name then lookup := t^.val
  else if name < t^.val.getName()  then lookup := lookup(t^.left, name)
  else if name > t^.val.getName()  then lookup := lookup(t^.right, name);
end;

begin
end.
```

Uwe Grohnwaldt
Yichen Dong
Johannes Rössel

A3.pas

```
{Aufgabenstellung: Programmieren Sie die Verwaltung der CDs in einem
 Hauptprogramm, in dem CDs in einer Baumstruktur abgelegt sind. Es sollen
 spezielle CDs gesucht werden koennen und das Gesamtverzeichnis nach Namen
 sortiert angezeigt werden koennen.}

program A3;

uses tree, UCd;

var t: Baum;

procedure Eintragen();
var cd: TCd;
    a: integer;
begin
  writeln('Moechten Sie eine');
  writeln('  1. Audio-CD');
  writeln('  2. Video-CD');
  writeln('eintragen?');
  write('-> ');
  readln(a);
  case a of
    1:begin
        cd := TAudioCd.Create();
      end;
    2:begin
        cd := TVideoCd.Create();
      end;
  end;
  cd.Input();
  writeln;
  Insert(t, cd);
end;

procedure suchen();
var s: String;
    c: TCd;
begin
  write('Bitte Namen der CD eingeben: ');
  readln(s);
  writeln;
  c := lookup(t, s);
  if c <> nil then begin
    writeln('Hab hier was tolles gefunden:');
    c.Output();
  end else writeln('Nix vorhanden hier, was dem entspricht');
  writeln;
end;

procedure Ausgabe(c: TCd);
begin
  writeln(c.getName());
end;

procedure table();
begin
  writeln;
  Traverse(t, @Ausgabe);
  writeln;
end;

function Menue():boolean;
var x: integer;
begin
  writeln('1. CD eintragen');
  writeln('2. CD suchen');
  writeln('3. Gesamtverzeichnis anzeigen');
```

```
    writeln('4. Programm beenden');
    write('-> ');
    readln(x);
    Menue := (x <> 4);
    case x of
      1: Eintragen();
      2: suchen();
      3: table();
    end;
  end;

  begin
    t := nil;
    while Menue do begin end;
    writeln('Bye.');
  end.
```