BWINF
Aufgabe: Bücherregal

```
int Zahl_Textdatei = 1;
String Anfang_Textdatei = "buecherregal";
int Anzahl_Deko_Figuren;
int Anzahl_Buecher;
IntList Hoehe_Buecher = new IntList();
IntList Hoehe_Buecher_sortiert = new IntList();

ArrayList<IntList> Abschnitt = new ArrayList<IntList>();
```

Hier werden die Variabeln deklariert und mache direkt initialisiert.

```
void setup() {
  ladeTextdokument(Anfang_Textdatei+Zahl_Textdatei+".txt", true);

  for (int i = 0; i< Anzahl_Deko_Figuren+1; i++) {
    Abschnitt.add(new IntList());
  }

  sortiere_Buecher(true);
  ordneZu();
  pruefeObMoeglich();
}
void draw() {
}
```

void setup() wird automatisch einmal beim Programmstart ausgeführt. void draw() wird jeden frame wieder ausgeführt. Da keine befehle ständig wiederholt werden müssen ist void draw leer. void setup startet alle (weiter unter deklarierten) Funktionen zum Einlesen des Textes , sortieren nach Buchhöhe , einteilen in gruppen und Prüfung ob alle bücher eingeteilt werden konnten.

```
void ladeTextdokument(String dateiname, boolean output) {
  if (output) {
    println("Lese Daten aus Textdatei" + dateiname);
    println();
  }
  Anzahl_Deko_Figuren = Integer.parseInt(loadStrings(dateiname)[0]);
  if (output) {
    println("Anzahl Deko-Figuren = " + Anzahl_Deko_Figuren);
  }

  Anzahl_Buecher = Integer.parseInt(loadStrings(dateiname)[1]);
  if (output) {
    println("Anzahl Bücher = " + Anzahl_Buecher);
    print("Höhe Bücher : ");
  }
  for (int i = 0; i < Anzahl_Buecher; i++) {
    Hoehe_Buecher.append(Integer.parseInt(loadStrings(dateiname)[i+2]));
    if (output) {
      print(Hoehe_Buecher.get(i)+" , ");
    }
  }
  if (output) {
    println();
  }
}
```

In der Funktion “ladeTextdokument” wird das textdokument eingelesen und in die Variabeln /undArray Lists geschrieben. Die Variabel output bestimmt ob die funktion ausgaben in die konsole machen soll

```
void sortiere_Buecher(boolean output) {
  Hoehe_Buecher_sortiert = Hoehe_Buecher;
  Hoehe_Buecher_sortiert.sort();
  if (output) {
    print("Höhe Bücher sortiert : ");
    for (int i = 0; i < Hoehe_Buecher_sortiert.size(); i++) {
      print(Hoehe_Buecher_sortiert.get(i)+" , ");
    }
    println();
  }
}
```

In der Funktion “sortiere_Buecher” werden die bücherhöhen (die in der IntList “Hoehe_Buecher” gespeichert sind) sortiert in der IntList “Hoehe_Buecher_sortiert” abgelegt.

```
void ordneZu() {
  println("teile Daten in Abschnitte ein");
  println();
  int counter = 0;

  for (int i = 0; i<Anzahl_Deko_Figuren+1; i++) {
    while ((counter < Hoehe_Buecher_sortiert.size()) && pruefeHoehenunterschied(i, counter)) {
      Abschnitt.get(i).append(Hoehe_Buecher_sortiert.get(counter));
      counter ++;
    }
    println("Abschitt "+(i+1)+" : "+Abschnitt.get(i));
  }
}

boolean pruefeHoehenunterschied(int i, int counter) {
  if (Abschnitt.get(i).size() > 0) {
      return Hoehe_Buecher_sortiert.get(counter) - Abschnitt.get(i).min() < 30;
  } else {
    return true;
  }
}
```

In der Funktion “ordneZu” werden die höhen in Gruppen eingeteilt. Diese Funktion greift auf die Funktion “pruefeHoehenunterschied” zu.

```
 void pruefeObMoeglich() {
   int summe = 0;
   for (int i = 0; i<Anzahl_Deko_Figuren+1; i++) {
     summe += Abschnitt.get(i).size();
   }
   if (summe < Anzahl_Buecher) {
     println("Nicht möglich");
   } else {
     println("möglich");
   }
 }
```

Die Methode “pruefeObMoeglich” addiert die Anzahl aller eingeteilten bücher mit der wirklichen Bücheranzahl. Stimmt diese überein konnte die Aufgabe gelöst werden.

```
Lese Daten aus Textdateibuecherregal1.txt

Anzahl Deko-Figuren = 4
Anzahl Bücher = 11
Höhe Bücher : 168 , 170 , 202 , 211 , 229 , 233 , 254 , 260 , 272 , 306 , 307 ,
Sortiere Daten

Höhe Bücher sortiert : 168 , 170 , 202 , 211 , 229 , 233 , 254 , 260 , 272 , 306 , 307 ,
teile Daten in Abschnitte ein

Abschitt 1 : IntList size=2 [ 168, 170 ]
Abschitt 2 : IntList size=3 [ 202, 211, 229 ]
Abschitt 3 : IntList size=3 [ 233, 254, 260 ]
Abschitt 4 : IntList size=1 [ 272 ]
Abschitt 5 : IntList size=2 [ 306, 307 ]
möglich
```

Eine typische Programmausgabe im Terminal