Programmierung in C++: Doppelt verkettete Liste, Sortierverfahren

- Erstellen Sie eine Klasse "**DVKE**" zur Verwaltung von Elementen einer doppelt verketteten Liste. "DVKE" habe als Attribute die Pointer auf das Vorgängerelement ("V") und auf den Nachfolger ("N"). Für Daten steht das Attribut void *Data zur Verfügung. Geben Sie diese Attribute (Zugriffsschutz: protected), den Initialisierungskonstruktor (Initialisierungsliste!) und die get- und set-Methoden der Attribute an.

- Für die Speicherung von geografischen Koordinaten mit Breiten- und Längengrad-angaben jeweils mit Grad, Minuten und Sekunden dient die Klasse "**GEOKO**". "GEOKO" hat die Attribute "int BrGr", "int LaGr", "int BrMin", "int LaMin", "double BrSec"und "double LaSec". Alle Attribute sind private.
  Geben Sie den Initialisierungskonstruktor an bei Verwendung der Initialisierungsliste und die get- und set-Methoden der Attribute. Überladen Sie die zum Sortieren erforderlichen Vergleichsoperatoren der Klasse "GEOKO"!

- Erstellen Sie eine Klasse "**DVK**", in der geografische Koordinaten mit Hilfe von Objekten der Klasse "DVKE" als doppelt verkettete Liste je mit einem Vorgänger und einem Nachfolger gespeichert werden können. Dazu **erbt** "DVK" von "DVKE"!
  Ein Attribut mit dem Namen "Max" gibt die maximale Anzahl der Listenelemente an, das Attribut "Anz" die tatsächliche Anzahl . Alle Attribute sind oder werden(!) private.

  Die geerbten Attribute erhalten jetzt die Bedeutung: Attribut "N" als Anker für die "Vorwärts-Verkettung" und "V" als Anker für die "Rückwärts-Verkettung" der doppelt verketteten Liste. "Data" wird für ein "GEOKO"-Objekt verwendet, in dem der Mittelwert der Längen- und Breitengradangaben aller Elemente der Liste gespeichert wird. Diese Mittelwerte sind bei jeder Erweiterung der Liste neu zuberechnen.
  In einem zusätzlichen Attribut vom Typ **Array von Pointern** auf "DVKE" mit dem Namen "**Index**" sind die Pointer auf die zugehörigen Listen-Elemente gespeichert. Dieses Array ist dynamisch anzulegen, die Anzahl der "DVKE"-Objekte (=Größe des Arrays) ist im Konstruktor zu übergeben und im Attribut "**Max**" zu speichern. Legen Sie hierzu den Konstruktor mit der Signatur DVK (int Anzahl) an, der eine leere verkettete Liste anlegt für maximal "Anzahl" Objekte, verwenden Sie auch hier die Initialisierungsliste!

  Erstellen Sie die Methode "**anhaenge**" mit der Signatur bool anhaenge(GEOKO *). Diese Methode fügt ein neues "DVKE"-Objekt mit dem übergebenen "GEOKO"-Objekt als Daten am Ende der Liste bezüglich der Vorwärtsverkettung ein und passt die Mittelwerte für das geerbte Attribut "Data" an. Stellen Sie die Verkettung durch geeignete Werte für die Attribute und die "V"- und "N"-Werte der "DVKE"-Objekte her und legen Sie die Adressen der Knoten zusätzlich im Array "Index" ab.

  Weiterhin ist in "DVK" eine Methode bereit zu stellen, welche die Datenobjekte bezüglich des Abstandes der einzelnen Daten vom "Mittelpunkt" (=Attribut "Data") sortiert. Programmieren Sie hierfür die Methode void heapSort() für das Heap-Sort-Verfahren. Messen Sie die Zeit, die für den Sortiervorgang gebraucht wird in Mikrosekunden.
  Sortieren Sie das Array "Index" und beachten Sie, dass beim Sortieren die Verkettung der doppelt verketteten Liste erhalten bleiben muss! Der Arrayindex entspricht für die Sortieralgorithmen der Position - 1 innerhalb der Liste.

Zum Vergleich der Sortiergeschwindigkeit ist in "DVK" eine weitere Sortier-Methode bereit zu stellen, die ebenso die Datenobjekte bezüglich des Abstandes vom "Mittelpunkt" (=Attribut "Data") der einzelnen Daten sortiert. Programmieren Sie hierfür die Methode void bubbleSort() für das Bubble-Sort-Verfahren. Messen Sie auch hier die Zeit, die für den Sortiervorgang gebraucht wird in Mikrosekunden. Sortieren Sie das Array "Index" und beachten Sie auch hier, dass beim Sortieren die Verkettung der doppelt verketteten Liste erhalten bleiben muss! Auch hier entspricht der Arrayindex für die Sortieralgorithmen der Position - 1 innerhalb der Liste.

- Erstellen Sie ein Hauptprogramm zur Anwendung der doppelt verketteten Liste. Legen Sie ein Menu an mit den Einträgen
  1.) Verkettete Liste anlegen,
  2.) Liste sortieren mit "Heap Sort",
  3.) Liste sortieren mit, "Bubble Sort"
  4.) Daten der Liste schreiben
  5.) Ende.

  Bei 1.) erfragen Sie vom Benutzer die Anzahl der Daten für die verkettete Liste und legen mit dieser Information ein "DVK"-Objekt an. Erfragen Sie weiterhin vom Benutzer den Dateinamen der Daten, lesen Sie diese Daten ein und erweitern Sie damit die erzeugte Liste! Bereitgestellt sind in Ilias die Dateien "Daten1.csv" und "Daten2.csv". Die in den Dateien angegebenen Werte sind Geo-Koordinaten jeweils in Sekunden mit zwei Nachkommastellen. Ermitteln Sie daraus die Werte für die Attribute (geeigneter Konstruktor???).
  Geben Sie die Daten des Mittelpunktes der fertigen Liste aus, in dem Sie eine geeignete Methode für DVK erstellen und benutzen!

  Bei 4.) sind die Daten in eine neue csv-Datei zu schreiben, der Name entspricht dem der eingelesenen Datei mit angehängtem "_S" (z.B. Daten1_S.csv). In der ersten Zeile der Datei stehen die Anzahl der Daten und die Koordinaten des Mittelpunktes. Danach folgen die sortierten Koordinaten jeweils eine Koordinate pro Zeile. Neben den Geo-Koordinaten ist auch der jeweilige Abstand zum Mittelpunkt anzugeben!

Bemerkung: Zur Verwendung des Attributes "Data" für GEOKO-Objekte verwenden Sie geeignete cast-Operatoren.

Bringen Sie für Ihre Vorführung sowohl den fertigen, lauffähigen Code als auch die Daten (csv-Dateien) in maschinenlesbarer Form (z.B. Memorystick) mit und führen Sie Ihr Programm in der Entwicklungsumgebung "CodeBlocks" auf den Laborrechnern vor. Beachten Sie bezüglich der Zeitmessung, dass die von Windows bereitgestellten Methoden unter Linux (Labor-Umgebung!) nicht vorhanden sind!

Viel Spaß beim Programmieren und debuggen!

Abgabe: 18. Januar 2019