Prüfung 2018

Aufgabe 1

a) *Zunächst soll eine abstrakte Klasse* ***Schaltungselement*** *mit folgenden Eigenschaften definiert werden:*

- *Die Klasse soll zwei private Attribute* ***x*** *und* ***y*** *Type double besitzen, die festlegen an welchem Punkt sich ein Schaltungselement innerhalb eines Koordinatensystems im Layout befindet.*
- *Die Klasse soll einen parameterlosen, protected Konstruktor besitzen, der die Attribute* ***x*** *und* ***y*** *jeweils mit 0 initialisiert.*

- *Die Klasse soll einen Public Konstruktor mit zwei Parametern* ***x*** *und* ***y*** *von Typ double besitzen, der die Parameterwerte in die gleichnamigen Attribute kopiert.*

- *Die Klasse soll eine public Methode ändern mit Ergebnistyp void und zwei Parametern* ***x_new*** *und* ***y_new*** *vom Typ double besitzen. Das Attribut x soll dem Wert von* ***x-new*** *bekommen und das Attribut y den Wert von* ***y_new****.*

- *Die Klasse soll eine Methode* ***get_x_y*** *mit Ergebnistyp void besitzen* ***get_x_y*** *soll die Werte der Attribute x und y über Referenzparameter nach außen geben, d.h* ***get_x_y*** *hat zwei Parameter vom Typ Referenz auf double, denen die Werte den Attribute* ***x*** *und* ***y*** *zugewiesen werden.*

- *Die Klasse soll eine abtstrakte, parameterlose Methode ausgabe mit Ergebnistyp void besitzen. Die Methode wird in Unterklassen implementiert*

b) *Von der Klasse Schaltungselement soll nun eine Klasse Kondensator als Unterklasse mit public Vererbung abgeleitet werden mit folgenden Eigenschaften.*

- *Die Klasse Kondensator besitzt ein zusätzliches privates Attribut **kapa** von Typ double für die Kapazität eines Kondensators.*
- *Die Klasse Kondensator soll einen Öffentlichen Konstruktor haben mit drei Parametern x, y und k, alle vom Typ double, mit x und y für die x und y-Koordinaten eines Kondensators und k für die Kapazität. Die Parameterwerte sollen auf geeignete Art und Weise an die Attribute x, y und kapa übergeben werden*
- *Die Klasse Kondensator soll eine Implementierung der abstrakten Methode ausgabe der Oberklasse besitzen, die die Daten eines Kondensators in folgender Form hier für einen Kondensator mit einem Kapazitätswert von 200 und x-Koordinaten 1,5 und y-Koordinate 2.7) inkl. Kopfzeile“ Kondensator“ ausgibt.*

Außer dem Zeilenwechsel müssen dabei keine weiteren Ausgabeformatierungen (ZB. Anzahl Nachkommerzahlen) gemacht werden.

Beachten Sie dabei, dass der direkte Zugriff auf die ererbten, privaten Attribute x und y nicht möglich ist!

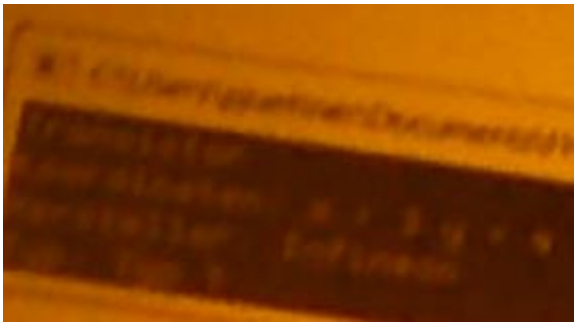
c) *Von Klasse Schaltungselement soll eine Klasse Transistor als Unterklasse mit public Vererbung abgeleitet werden mit folgenden Eigenschaften:*

- *Die Klasse Transistor soll die privaten Attribute Hersteller und Typ beide vom Typ char besitzen. Dabei sollen der pointer Hersteller auf eine auf dem Heap befindliche Zeichenkette mit dem Herstellernamen und der pointer Typ auf eine dem Heap befindliche Zeichenkette mit dem Typnamen des Transistors zeigen.*

- *Die Klasse Transistor soll einen Konstruktor mit vier Parametern für die x- und y- Koordinaten, sowie für Hersteller und Typ besitzen. Die Parameterwerte, für die x- und y-Koordinaten sollen in die entsprechenden Attribute geeignet übertragen werden. Die Zeichenketten, auf die die Parameter für Hersteller und Typ zeigen, sollen auf dem Heap kopiert werden. Die Attribute für Hersteller und Typ sollen dann auf die entsprechenden neu angelegten Heapbereiche zeigen.*

Verwenden Sie für Speicherplatzanforderung nur den Operator new angewandte auf Vektoren von Typ char. Fehlerabfragen müssen nicht programmiert werden.

- *Die Klasse Transistor soll eine Implementierung der abstrakten Methode folgender Form (hier für einen Transistor mit Hersteller Infineon. Typ Typ 1 und x-Koordinate 3 und y-Koordinaten 4) inklusive Kopfzeile. Transistor ausgibt.*



Außer dem Zeilenwechsel müssen dabei keine weiteren Ausgabeformatierungen (ZB. Anzahl Nachkommerzahlen) gemacht werden

- *Die Klasse Transistor soll einen Kopierkonstruktor besitzen, der ein Objekt der Klasse Transistor kopiert. Dabei sollen die auf dem Heap befindlichen Zeichenketten auf dem Heap kopiert werden Zur Ermittlung der x- und y-Koordinaten den zu kopierenden Objekts soll die Oberklassenmethode **get_x_y** geeignet*

*aufgerufen werden. Zum Setzen der **x-** und -**y** Koordinaten des Zielobjekts soll die Oberklassenmethode ändern geeignet aufgerufen werden. Ein Oberklassen Konstruktor muss hier nicht explizit aufgerufen werden, d.h es kann der parameterlose Konstruktor der Oberklasse implizit verwendet werden. Fehlerabfragen müssen nicht programmiert werden. Beachten Sie die notwendige Signatur des Kopierkonstruktors.*

- *Die Klasse Transistor soll einen Destruktor besitzen, der den belegten Speicherplatz Bereich weder freigibt. Fehlerabfragen müssen dabei nicht programmiert werden.*

d) Schreiben Sie ein Hauptprogramm, in dem ein Kondensator Objekt mit den x- und y- Koordinaten 1.5 und 2.7 und einer Kapazität von 200 auf dem Heap angelegt wird. Die Adresse des Objekts soll in einem Pointer auf Schaltungselement abgespeichert werden. Anschließend sollen die Daten des Kondensators mittels der Methode ausgabe auf dem Bildschirm ausgegeben werden.

C) Erläutern Sie warum in den Klassen Schaltungselement, Kondensator und Transistor keine Memberinitialiesierung notwendig ist.

2

a) Schreiben Sie für die Klasse Matrix einen überladenen Ausgabeoperator << der die Signatur des Standardausgabeoperators angepasst übernimmt und der die Elemente einer Matrix zeilenweise und durch ein Leerzeichen getrennt ausgibt.

D.h , Zwischen zwei Elementen der Matrix soll ein Leerzeichnen ausgegeben werden und nach jeder Zeile soll ein Zeilenwechsel stattfinden Beispielhaft ist die Ausgabe für eine 2 x 2 Matrix mit den Elementen 1 und 2 in der ersten Zeile und 3 und 4 in der zweiten Zeile unten dargestellt. Weiters Ausgabeformatierungen sind nicht notwendig Ergänzen Sie die Klasse Matrix oben wo erforderlich

b) Erläutern Sie warum friend -Funktion nicht dynamisch gebunden werden können.