01. Programmierung in C++

In diesem Praktikum ist ein Programm zu entwickeln, um die Lage von Kreisen zueinander bestimmen zu können. Die Kreise sollen durch zwei verschiedene Möglichkeiten angegeben werden können, zum einen durch ihren Mittelpunkt und einen Radius, zum anderen durch Konstruktion aus drei Punkten. Halten Sie die folgende Vorgehensweise ein:

Entwerfen Sie die Klassen POINT für Punkte in der kartesischen Ebene, LINE um Geraden darstellen zu können und CIRCLE für die Darstellung von Kreisen.

POINT hat die privaten Attribute double X und double Y, einen Initialisierungskonstruktor, einen Copy-Konstruktor, die Methode „distanceTo“ für den Abstand zweier Punkte und die get- und set-Methoden für die Attribute.

LINE **erbt** von POINT einen Punkt auf der Geraden, braucht also einen weiteren als Attribut, um die Gerade zu bestimmen. Verwenden Sie auch hier den Initialisierungs-konstruktor mit der Initialisierungsliste für die Attribute und die geerbte Klasse.

Auch CIRCLE erbt von POINT, dies soll der Mittelpunkt eines Kreis-Objektes sein. Als weiteres Attribut ist der Radius erforderlich. Verwenden Sie den Initialisierungskonstruktor und den Copy-Konstruktor unter Anwendung der Initialisierungsliste! Erstellen Sie die Methode „isInCircle“ um festzustellen, ob ein in der Parameterliste übergebener Punkt innerhalb des Kreises liegt und die Methode „meetsOther“, die für einen in der Parameterliste übergebenen weiteren Kreis die Anzahl der Schnittpunkte zurückgibt (zweiter Parameter) und gegebenenfalls Pointer auf diese Schnittpunkte! Beachten Sie, dass MeetsOther auch für den Schnitt von Kreisen verwendet werden kann, die unterschiedliche Radien aufweisen!

Verwenden Sie die in der Übung vorgestellten Formeln u.a. für die Ermittlung der Schnittpunkte. Beachten Sie, dass für die Rückgabe der Schnittpunkte Pointer an das aufrufende Programm zurückgegeben werden, also gemäß den Regeln für Call by Reference in der Signatur für die Pointer der Schnittpunkte der Datentyp POINT * * verwendet werden muss!

Entfernen Sie alle Objekte von Hilfskreisen und Hilfsgeraden durch den delete-Operator.

Entwickeln Sie auf der Grundlage der Klassen POINT, LINE und CIRCLE ein Programm mit dem folgenden Menu:

1.) Zwei beliebige Kreise je mit Mittelpunkt und Radius angeben und zum Schnitt bringen. Verhältnis der Kreise zu einander angeben und gegebenenfalls Berührpunkt oder Schnittpunkte anzeigen.

2.) Zwei Kreise mit Hilfe von jeweils drei beliebigen Punkten konstruieren und zum Schnitt bringen. Verhältnis der Kreise zu einander angeben und gegebenenfalls Berührpunkt oder Schnittpunkte anzeigen.

9.) Programmende

Anmerkung: Die Konstruktion eines Kreises mit Hilfe von drei Punkten sei wie folgt:

Mit drei Punkten der x-y-Ebene kann ein Kreis konstruiert werden, auf dessen Umfang diese drei Punkte liegen. Verwenden Sie für die Konstruktion von Kreisen die folgende Vorgabe:

1. Einlesen der Koordinaten der Punkte A, B, C und Erstellen der POINT-Objekte;
2. Fehlermeldung und Rückkehr in das Menu, falls zwei der Punkte gleich sind;
3. Konstruktion von **vier** Hilfskreisen mit den Mittelpunkten A, B, C so, dass sich je zwei benachbarte Kreise mit gleichem Radius schneiden, z.B je einen Kreis um A und B mit einem Radius Abstand A_B, und je einen Kreis um B und C mit einem Radius Abstand B_C;
4. Ermittlung der Geraden, die durch die Schnittpunkte der benachbarten Kreise definiert sind;
5. Fehlermeldung und Rückkehr in das Menu, falls die beiden ermittelten Geraden parallel sind. Das würde bedeuten, dass die verwendeten Punkte A, B und C auf einer Geraden liegen.
6. Bestimmung des Schnittpunktes der Geraden als Mittelpunkt des zu konstruierenden Kreises;
7. Ermittlung des Radius als Abstand zwischen Mittelpunkt und einem der Punkte A,B oder C und Erstellen eines CIRCLE-Objektes damit.

Verwenden Sie den Code der bereitgestellten Dateien und erstellen Sie damit ein Code::Blocksprojekt. Die Signaturen der angebenen Methoden dürfen nicht verändert werden, weitere Methoden dürfen Sie erstellen (zum Beispiel um Zwischenergebnisse zu ermitteln…)

Verwenden Sie die Code::Blocks-Version 20.03 mit dem Compiler gcc8.1.0 für Windows, unicode, 32 Bit.

Dokumentieren Sie die Funktionsfähigkeit Ihres Projektes durch Screen-Shots mit den folgenden Werten: Verwenden Sie die Quersumme modulo 9 der letzten 4 Ziffern Ihrer 8-stelligen Matrikelnummer als X-Koordinate des Mittelpunktes eines Kreises K1, die Quersumme modulo 9 der ersten 4 Ziffern Ihrer 8-stelligen Matrikelnummer als Y-Koordinate des Mittelpunktes des Kreises K1. Für einen weiteren Kreis K2 verwenden Sie analog die Matrikelnummer des zweiten Gruppenteilnehmers. Der Radius für K1 sei der Abstand der Mittelpunkte der beiden Kreise, der Radius für K2 sei 75% des Radius von Kreis 1.

Beispiel:

Kandidat 1, Matr.-Nr. 11092345: Quersumme der letzten 4 Ziffern = 14, 14 mod 9 = 5; Quersumme der ersten 4 Ziffern = 11, 11 mod 9 = 2, -> Mittelpunkt Kreis 1: (5|2)

Kandidat 2, Matr.-Nr. 11135678: Quersumme der letzten 4 Ziffern = 26, 26 mod 9 = 8 Quersumme der ersten 4 Ziffern = 6, 6 mod 9= 6;-> Mittelpunkt Kreis 1: (8|6)

Abstand der beiden Mittelpunkte: 5, gewählte Radien 5 und 3,75.

Erstellen Sie für die Abgabe eine gezippte Version Ihres Code::Blocks-Projektes und legen Sie dies und die Screenshots als jpg-Dateien in Ihrem Gruppenordner ab.

Viel Spaß bei Programmieren und entwanzen Ihres Programmes!

Abgabe: 11. 06. 2023