

OOP – Hausaufgabe 01

Thema: Geometrieklassen

Aufbau eines einfachen Systems elementarer Klassen für ebene geometrische Gebilde, Implementierung eines vorgegebenen Klassenentwurfs.

Vorbereitungen

- Legen Sie ein Projekt geometry (als leeres Projekt ohne vorkompilierte Header) an.

Teilaufgaben

Nr.	Beschreibung	Punkte
1	<p>Vorgegeben ist ein Entwurf folgender Klassen der Ebene:</p> <ul style="list-style-type: none"> • Vector2D (Vektor im \mathbb{R}^2), • Point2D (Punkt), • Line2D (Strecke), • Polyline2D („Knickstrecke“), • BoundingBox2D (achsenparalleles Umrandungsrechteck), <p>Lassen Sie dabei Polyline2D von Line2D erben. Implementieren Sie geeignete Konstruktoren, ggf. die Beziehungen zwischen den Klassen und erforderlichenfalls Destruktoren. Klasse Point2D soll über einen geeigneten Konstruktor auch ein ungültiges Objekt hervorbringen können, das über die Methode isValid() identifiziert werden kann (Wert false, s. u.). Damit soll das Problem gelöst werden können, unter bestimmten Bedingungen aus Methoden keinen gültigen Punkt zurückgeben zu können.</p>	10
2	<p>Implementieren Sie folgende Methoden der Klasse Vector2D:</p> <ul style="list-style-type: none"> • <code>double getX() const;</code> • <code>double getY() const;</code> • <code>Vector2D add(const Vector2D& other) const;</code> • <code>Vector2D subtract(const Vector2D& other) const;</code> • <code>double getLength() const;</code> • <code>double getOrientation() const;</code> • <code>double multiply(const Vector2D& other) const;</code> 	6
3	<p>Implementieren Sie folgende Methoden der Klasse Point2D:</p> <ul style="list-style-type: none"> • <code>double getX() const;</code> • <code>double getY() const;</code> • <code>bool isValid() const;</code> (siehe Teilaufgabe 1) • <code>Point2D add(const Vector2D& vect) const;</code> • <code>Vector2D getDifference(const Point2D& other) const;</code> • <code>double getDistance() const;</code> • <code>double getDistance(const Point2D& other) const;</code> • <code>void moveBy(const Vector2D& delta);</code> • <code>void moveBy(double deltaX, double deltaY);</code> • <code>void moveTo(double newX, double newY);</code> • <code>BoundingBox2D getBounds() const;</code> 	10

Nr.	Beschreibung	Punkte
4	<p>Implementieren Sie folgende Methoden der Klasse Line2D (als Strecke zu interpretieren):</p> <ul style="list-style-type: none"> • <code>Point2D getStart() const;</code> • <code>Point2D getEnd() const;</code> • <code>Point2D getPointAt(double offset) const;</code> • <code>double getLength() const;</code> • <code>Line2D add(const Vector2D& vect) const;</code> (verschiebt Endpunkt) • <code>Vector2D getVector() const;</code> (von Start- zu Endpunkt) • <code>void moveBy(const Vector2D& offs);</code> • <code>void moveBy(double deltaX, double deltaY);</code> • <code>double getDistance(const Line2D& other) const;</code> • <code>double getDistance(const Point2D& other) const;</code> • <code>BoundingBox2D getBounds() const;</code> 	10
5	<p>Modifizieren Sie die von Klasse Line2D ererbten Methoden in Klasse PolyLine2D, soweit nötig (deklarieren Sie diese Methoden in Line2D als virtual, Methode <code>add</code> hängt jetzt ein zusätzliches Liniensegment hinten an), und implementieren Sie zusätzlich folgende Methoden der Klasse PolyLine2D:</p> <ul style="list-style-type: none"> • <code>Point2D getPoint(unsigned short number) const;</code> • <code>unsigned short getNumberOfPoints() const;</code> • <code>bool insertPoint(const Point2D& pt, unsigned short before);</code> • <code>bool removePoint(unsigned short number);</code> • <code>bool isSelfIntersecting() const;</code> 	12
6	<p>Implementieren Sie für Klasse BoundingBox2D folgende Methoden:</p> <ul style="list-style-type: none"> • <code>double getWidth() const;</code> • <code>double getHeight() const;</code> • <code>void uniteWith(const BoundingBox2D& other);</code> 	5
7*	<p>Freiwillige Zusatzaufgabe</p> <p>Entwerfen Sie eine abstrakte Klasse Drawer2D, mit deren Hilfe jedes Objekt der in Teilaufgabe 1 umrissenen Klassen (außer Vector2D) sich in einer geeigneten Umgebung zeichnen könnte. Überlegen Sie sich dazu, welche elementaren Methoden die Klasse anbieten müsste. Implementieren Sie auf allen Klassen aus Teilaufgabe 1 (außer Vector2D) eine Methode draw, die das Objekt zeichnet:</p> <ul style="list-style-type: none"> • <code>void draw(Drawer2D& drawer) const;</code> 	7
8*	<p>Freiwillige Zusatzaufgabe</p> <p>Implementieren Sie eine instanziiierbare Unterklasse von Drawer2D aus Teilaufgabe 7, die mit Hilfe der Turtleizer_CPP-Bibliothek (https://github.com/codemanyak/Turtleizer_CPP) die Drawer2D-Methoden implementiert und somit ein Zeichnen der Objekte aus Teilaufgabe 1 tatsächlich ermöglicht.</p>	10