

Kerncurriculum berufliches Gymnasium Praktische Informatik

Fach: Praktische Informatik

Umsetzungsbeispiel für die Qualifikationsphase (1)

Das nachfolgende Beispiel zeigt eine Möglichkeit der Umsetzung ausgewählter Aspekte des Themenfeldes "Implementierung von Klassen und Assoziationen" im Unterricht der Qualifikationsphase. Es veranschaulicht exemplarisch, in welcher Weise die Lernenden in der Auseinandersetzung mit einem Themenfeld Kompetenzen erwerben können, die auf das Erreichen ausgewählter Bildungsstandards für die allgemeine Hochschulreife am beruflichen Gymnasium zielen (Verknüpfung von Bildungsstandards und Themenfeldern unter einer Schwerpunktsetzung).

Das ausgewählte Beispiel verdeutlicht, inwiefern sich eine Bezugnahme sowohl auf die fachdidaktischen Grundlagen (Abschnitt 2.3, 2.4) als auch auf Bildungsstandards und Unterrichtsinhalte (Abschnitt 3.2, 3.3) im Einzelnen realisieren lässt – je nach unterrichtlichem Zusammenhang und Zuschnitt des Lernarrangements.

Kurshalbjahr:	Q1: Objektorientierte Softwareentwicklung (LK)
Themenfeld:	Q1.2 Implementierung von Klassen und Assoziationen
Kontext:	Nutzung von Polymorphie zur Verwaltung verschiedenartiger Objekte in einer Liste
Didaktische Funktion:	Entwicklung eines 2D-Zeichenprogramm mit den Elementen: Strecke, Dreieck, Rechteck
Bezug zu den Leitideen:	
<ul style="list-style-type: none"> – Information und Daten (L1): Aus den Informationen der grafischen Darstellung eines Zeichenprogrammes lassen sich die Daten der zugehörigen 2D-Elemente gewinnen. – Algorithmen (L2): Der zur Realisierung des Zeichenprogramms benötigte Aufbau eines Klassenkonzeptes über virtuelle Klassen mit Vererbung wird derart modelliert, dass mit Hilfe einer virtuellen Zeichenfunktion die Darstellung beliebiger Objekttypen möglich ist. – Softwaretechnik und Sprachen (L3): Anschließend wird das entworfene Klassenkonzept in der eingeführten Programmiersprache implementiert. – Informationssysteme (L4): Bei der Erweiterung der Software durch Eingabefenster für die verschiedenen Objekttypen wird die Trennung der Anwendungsschicht von der Datenhaltungsschicht thematisiert. 	
Problemstellung:	
<p>Anhand einer gegebenen Software zur Darstellung von 2D-Objekten und eines Quelltextauszuges (siehe Abbildung) soll ein Klassendiagramm zur Realisierung einer gleichwertigen Software entwickelt werden. Die Lernenden erhalten den Auftrag das Klassendiagramm mittels polymorpher Klassen so zu planen, dass alle Objekte in einer Liste verwaltet werden können. Weiterhin soll das Konzept so gestaltet sein, dass neue Klassen wie Parallelogramm, Raute usw. leicht eingebunden werden können.</p>	

Anschließend erhalten die Lernenden den Auftrag die Software zu implementieren. Auf die Eingabe der Objekte durch den Anwender kann zunächst verzichtet werden, indem im Konstruktor der Klasse CZeichenprogrammDlg die Objekte wie folgt angelegt werden:

```

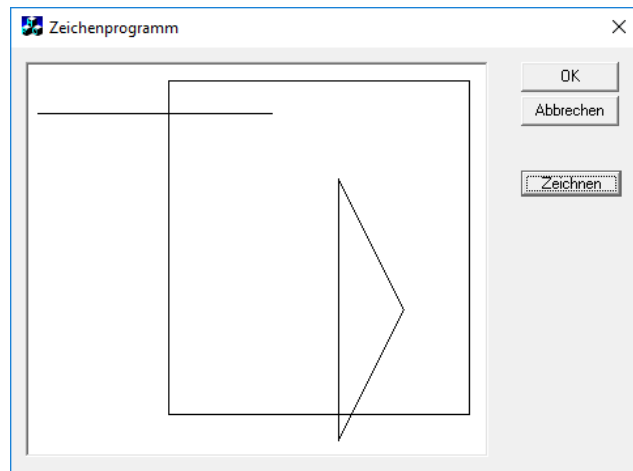
CElement* p_E;
std::list<CElement*> elemente;

CVektor v1(20,50), v2(200,50);
p_E = new CStrecke(v1,v2);
elemente.push_front(p_E);

CVektor v3(120,25), v4(350,280);
p_E = new CRechteck(v3,v4);
elemente.push_front(p_E);

CVektor v5(250,100), v6(250,300);
CVektor v7(300,200);
p_E = new CDreieck(v5,v6,v7);
elemente.push_front(p_E);

```



Kompetenzbereiche / Bildungsstandards

- Kommunizieren und Kooperieren (K1)
- Analysieren und Interpretieren (K2)
- Entwickeln und Modellieren (K3)
- Entscheiden und Implementieren (K4)
- Reflektieren und Beurteilen (K5)

Lernaktivitäten

Die Lernenden

- analysieren das Programm bezüglich der gegebenen Funktionalität (K1, K2),
- entwickeln ein UML-Klassendiagramm (K3),
- implementieren die Software anhand des UML-Klassendiagramms (K4),
- beurteilen die Erweiterungsmöglichkeiten ihres Softwarekonzeptes (K5).

Mögliche Vertiefungen können z.B. die flexible Eingabe von Objekten über die GUI sowie die Bestimmung von Schnittpunkten der Objekte sein (K4).

Materialien / Literatur / Links:

