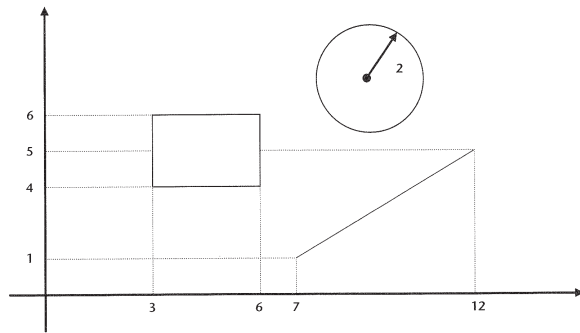
**Durchführung:**

Setzen Sie das Klassendiagramm in C++ um. Beachten Sie dabei die modulare Programmgestaltung.

Kontrolle:

Schreiben Sie ein Test-Hauptprogramm, in dem alle Funktionalitäten der Klassen und deren Beziehungen getestet werden können.

Beispielsweise sollen die folgenden geometrischen Grundfiguren erfasst werden (durch die Punkte im Koordinatensystem) und die korrekte Länge und die korrekten Flächen berechnet und angezeigt werden:



Fläche Rechteck: $3 \cdot 2 = 6 \text{ (LE}^2\text{)}$

LE....Längeneinheiten

Fläche Kreis: $\pi \cdot 2^2 \approx 12,57 \text{ (LE}^2\text{)}$

Länge Linie: $\sqrt{5^2 + 4^2} \approx 6,40 \text{ (LE}^2\text{)}$ (Pythagoras benutzen)

Lernziele:

- ▶ Sie lernen die wichtigen Aspekte der formalen Sprache UML kennen.
- ▶ Sie erarbeiten sich die Grundlagen für die Umsetzung von Klassen und deren Beziehungen in C++
- ▶ Sie erkennen den Nutzen eines CASE-Tools und dessen Einsatzgebiet.

Lernsituation 12:

Implementierung einer Klasse zur Simulation der echten Bruchrechnung

Ausgangssituation:

Neben der Entwicklung eines Geometrie-Programmes für den Einsatz im Mathematikunterricht der Oberstufe plant die Firma **ProSource** auch eine Software-Reihe für den Mathematikunterricht in der Grundschule und den Klassen der Sekundarstufe 1. Ein Teilmodul dieser Software soll ein Übungsprogramm für die Bruchrechnung werden. Dazu soll eine Klasse in C++ entwickelt werden, die den mathematischen Bruch korrekt repräsentiert. Dadurch soll die Grundlage für eine effiziente und fehlerfreie Entwicklung des Übungsprogrammes geschaffen werden.

Da Sie bereits erfolgreich die Klassen für das Geometrie-Programm implementiert haben, erhalten Sie den Auftrag, die Bruch-Klasse umzusetzen.

Arbeitsschritte in Einzel- oder Partnerarbeit:

Planung:

Die Bruch-Klasse soll einen mathematischen Bruch repräsentieren. Mathematisch gesehen ist ein Bruch der Quotient aus zwei ganzen Zahlen. Der Datentyp für eine ganze Zahl in C++ ist der vorzeichenbehaftete Integer (Beispiel in C++: `long int zaehler`).

Aus der Entwicklungsabteilung erhalten Sie eine Vorgabe der Bruch-Klasse, die von Ihnen implementiert und erweitert werden muss.

```
class CBruch
{
    private:
        long zaehler, nenner;

    public:
        CBruch();
        CBruch(long, long);
        CBruch(double);
        void Kuerze();
        :      // weitere Methoden und
        :      // überladene Operatoren
};
```

Die Konstruktoren der Klasse sind in der Lage sowohl Integer-Werte als auch Gleitpunktzahlen in einen Bruch umzuwandeln, wobei die entsprechenden Werte in den privaten Attributen `zaehler` und `nenner` gespeichert werden.

Das Kürzen des Bruches soll die Funktion `Kuerze()` übernehmen.

Beispiel:

```
#include <iostream>
using namespace std;

int main()
{
    CBruch A;                // repräsentiert: 0/1
    CBruch C(3.45);          // repräsentiert: 345/100
    CBruch D( 10 , 20);      // repräsentiert: 10/20
    D.Kuerze();              // repräsentiert: 1/2

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Weiterhin soll die Klasse über Methoden verfügen, die sie zur „echten“ Bruchrechnung befähigt. Analysieren Sie dazu auch die handschriftliche Skizze aus der Kontroll-Phase, die (neben der Vorlage für das Testen) Aufschluss über die Fähigkeiten der Klasse gibt.

Die Umsetzung des Bruch-Klasse setzt eine intensive Auseinandersetzung mit dem **Kapitel 12 (Überladen von Operatoren)** voraus.

Durchführung:

Implementieren Sie die Bruch-Klasse in C++. Benutzen Sie dazu vor allem die Technik der Operatoren-Überladung. Neben den arithmetischen Operatoren (+, -, *, /) sollen auch die Ein- und Ausgabe-Operatoren überladen werden, so dass eine komfortable Anzeige auf dem Bildschirm möglich ist. Diese Anzeige ist zwar in dem späteren Übungsprogramm nicht mehr nötig, da eine graphische Benutzeroberfläche für die Anzeige entwickelt wird, ist jedoch in der Entwicklungs- und Testphase sehr hilfreich.

Kontrolle:

Als Vorlage für das Testen der Bruch-Klasse dient eine handschriftliche Skizzierung, die von dem Leiter der Software-Entwicklung angefertigt wurde.

Schreiben Sie ein Test-Programm, in dem die Bruch-Klasse auf alle skizzierten Möglichkeiten getestet wird.

Skizzierung der Bruch-Klasse vom Leiter der Software-Entwicklung:

Bruch-Klasse:

1) Umwandlung von double: $\text{CBruch } A(1.23);$
 \uparrow
 $\frac{123}{100}$

2) Grundrechenarten:

$\text{CBruch } A(1,2); \rightsquigarrow A = \frac{1}{2}$
 $\text{CBruch } B(3,4); \rightsquigarrow B = \frac{3}{4}$
 $\text{CBruch } C;$

$C = A + B; \rightsquigarrow C = \frac{5}{4}$

$C = B - A; \rightsquigarrow C = \frac{1}{4}$

$C = A * B; \rightsquigarrow C = \frac{3}{8}$

$C = B / A; \rightsquigarrow C = \frac{6}{4} = \frac{3}{2}$

Brüche immer
vollständig
kürzen.

3) "Überladen": Typ double

$C = A + 1.5; \rightsquigarrow C = \frac{2}{1}$

$C = 3.75 - B; \rightsquigarrow C = \frac{3}{1}$

$C = 10 + B; \rightsquigarrow C = \frac{43}{4}$

\uparrow
Typ long

4) Ein- & Ausgabe

$\text{cout} \ll A;$

$\text{cout} \ll \text{"Bruch"} \ll A;$

$\text{cin} \gg A;$

$\text{cin} \gg A \gg B;$

Lernziele:

- Sie lernen eine wichtige Technik in C++ kennen – das Überladen von Operatoren.
- Sie erkennen den Nutzen dieser Technik für die realitätsnahe Umsetzung einer Software.