

## Musterklausur

### Aufgabe 1 - Gemischtes

Füllen Sie die Lücken aus oder kreuzen Sie an:

1. Polymorphe Klassen werden in C++ mit Hilfe von \_\_\_\_\_ Methoden implementiert.
2. Der Funktionsaufruf `foo(double &x)` nennt sich 'Call-by- \_\_\_\_\_'.
3. Eine Variable, die außerhalb von Funktionen und Klassen definiert wird, nennt sich \_\_\_\_\_ Variable.
4. Eine C++ Quelltextdatei wird beim Kompilieren in eine \_\_\_\_\_ -Datei übersetzt.
5. Methoden sind Implementierungen von \_\_\_\_\_.
6. In der Objektorientierung gibt es drei wichtige Prinzipien (Basiswerkzeuge):
  - \_\_\_\_\_
  - \_\_\_\_\_
  - \_\_\_\_\_
7. Eine Klasse ist der Bauplan eines \_\_\_\_\_.
8. Beschreiben Sie den Unterschied zwischen Daten und Eigenschaften (Attributen) in der Objektorientierung.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
9. Welche/r der folgenden Bezeichner in C++ sind/ist korrekt? Markieren Sie diese/n durch ein Kreuz (☒).
  - ☐ new
  - ☐ .hallo
  - ☐ 5h
  - ☐ c++
10. Das C++ Schlüsselwort \_\_\_\_\_ kann in Member-Funktionen von Klassen aufgerufen werden und gibt einen Zeiger auf das aufrufende Objekt dieser Klasse zurück.
11. Eine C++-Exception wird mit dem Schlüsselwort \_\_\_\_\_ ausgelöst.

### Aufgabe 2 - Modellierung

1. Was ist ein Modell? Nennen Sie 3 Beispiele aus dem Alltag und beschreiben Sie kurz, warum es sich bei den drei Beispielen um Modelle handelt.
2. Das Ausleihsystem (A) der FH-Bibliothek ermöglicht es einem\*einer Nutzer\*in (N) ein oder mehrere Bücher gleichzeitig auszuleihen. In der Bibliothek (Bib) stehen zwei Bücher (B1 und B2) zur Verfügung.

- (a) Erstellen Sie ein abstraktes Modell (Skizze) welches alle möglichen Zustände und deren Übergänge der Ausleihe beschreibt.
- (b) Überführen Sie die Objekte eines der oben modellierten Zustände in ein UML-Diagramm. Wählen Sie hierzu einen geeigneten Diagrammtyp und skizzieren Sie es. Machen Sie des weiteren die Beziehungen zwischen den Objekten im Diagramm deutlich.

### Aufgabe 3 - Operatorüberladung

1. Gegeben sei folgendes C++-17 Programm. Wie lautet die Ausgabe?

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

class Vektor{

public:
    Vektor() = default;
    Vektor(double x, double y): x{x}, y{y} {}
    Vektor operator+ (const Vektor &other)
    {
        Vektor out;
        out.x = this->x * other.x;
        out.y = this->y * other.y;
        return out;
    }

    Vektor operator* (const Vektor &other)
    {
        Vektor out;
        out.y = this->x + other.x;
        out.x = this->y + other.y;
        return out;
    }

    void print()
    {
        cout << x << " " << y << endl;
    }

private:
    double x {0.0};
    double y {0.0};

};

int main()
{
    Vektor result = Vektor(1.3, 2.0) + Vektor(1.4, 1.0) * Vektor(1.0, 1.0);
    result.print();
    return 0;
}
```

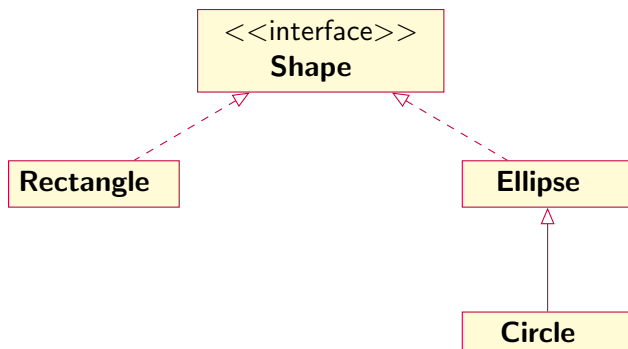
2. Der Klasse Vektor aus obigem Quelltext soll angepasst bzw. erweitert werden. Schreiben Sie zur Lösung der Aufgabe die Vektor-Klasse neu auf:
  - (a) Schreiben Sie die Klasse Vektor so um, dass durch die Verknüpfung zweier Vektoren mit dem Operator \* nun ein Skalarprodukt berechnet wird.

**Hinweis:** Die mathematische Definition des Skalarproduktes lautet:  $\vec{v}_1 \cdot \vec{v}_2 = x_1 x_2 + y_1 y_2$  mit  $\vec{v}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$  und  $\vec{v}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$

- (b) Wie ist die Klasse Vektor zu erweitern, so dass die main-Funktion aus obigem Quelltext ohne Änderung ausführbar bleibt?

## Aufgabe 4 - Objektorientierung

Gegeben sei folgendes Klassendiagramm:



1. Implementieren Sie die Klassen aus obigem Klassendiagramm so, dass die folgende main-Funktion ausgeführt werden kann:

```

#include "shape.hpp"
#include "circle.hpp"
#include "ellipse.hpp"
#include "rectangle.hpp"

int main()
{
    Circle circle;
    Rectangle rectangle;
    Ellipse ellipse;

    Shape* shapes[] = {&circle, &rectangle, &ellipse};

    for(auto shape : shapes)
    {
        shape->zeichnen();
    }

    return 0;
}

```

und folgende Ausgabe macht:

```

zeichne Ellipse
zeichne Rectangle
zeichne Ellipse

```

Schreiben Sie nur Quelltext, der für die Lösung der Aufgabe notwendig ist. (Für überflüssigen Quelltext werden Punkte abgezogen!)

2. Erweitern Sie das obige Klassendiagramm so, dass es möglichst genau Ihren Quelltext beschreibt.

3. Das System wird durch eine Display-Klasse erweitert, welche beliebig viele Shapes darstellen (zeichnen) können soll. Welche Assoziation beschreibt die Beziehung zwischen Shape und Display Klasse am besten? Skizzieren Sie diese in einem UML-Klassendiagramm und begründen Sie Ihre Auswahl.