

In den ersten beiden Aufgaben sollen Sie Speicherzustände zeichnen. Angenommen wir haben folgenden Java Code.

```

public class Main {
    public static void main(String[] args) {
        int x = 42;
        Wrapper w = new Wrapper();
        w.value = 0;
        f(w);
    }

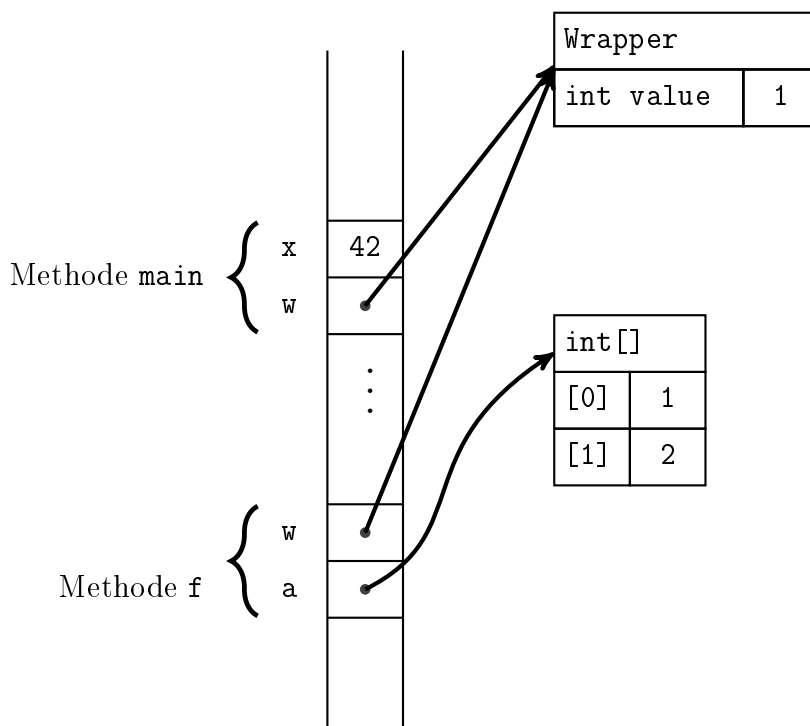
    public class Wrapper {
        int value;
    }

    public static void f(Wrapper w) {
        int[] a = {1,2};
        w.value = 1;

        // Speicherzustand hier gezeichnet
    }
}

```

Dann sieht der Speicher an der markierten Stelle wie folgt aus:



Aufgabe 2 (Seiteneffekte):

(2 + 2 + 2 = 6 Punkte)

Betrachten Sie das folgende Programm:

```

public class HSeiteneffekte {
    public static void main(String[] args) {
        Wrapper w1 = new Wrapper();
        Wrapper w2 = w1;

        w1.setI(1);
    }
}

```

```

    w2.setI(2);

    int x = 3;
    int[] a = { 1, 2 };

    f(w1, x, new int[] { 4, 5 });
    f(w2, x, a);
    //Speicherzustand hier zeichnen
}

public static void f(Wrapper w, int x, int[] a) {
    //Speicherzustand hier zeichnen
    x = a[0];
    a[0] = w.getI();
    w.setI(x);
}

}

public class Wrapper {
    private int i;

    public void setI(int x) {
        i = x;
    }
    public int getI() {
        return i;
    }
}

```

Es wird nun die Methode `main` ausgeführt. Stellen Sie den Speicher (d.h. alle (implizit) im Programm vorkommenden Arrays (außer `args`) und Objekte) bei jedem Aufruf der Methode `f` und vor Ende des Programms graphisch dar. Es müssen also insgesamt drei Speicherzustände dargestellt werden.

Aufgabe 4 (Einfache Klassen):

(2 + 2.5 + 1 + 2.5 = 8 Punkte)

- a) Schreiben Sie eine Klasse **Vector** mit einem Attribut, das ein Array vom Typ **double** referenziert. Ein Objekt vom Typ **Vector** soll einen Vektor im n -dimensionalen Raum repräsentieren. Erstellen Sie hierzu eine Methode **static Vector newWithDimension(int n)**, welche ein neues Objekt vom Typ **Vector** erzeugt und das Attribut mit einem Array der Größe n initialisiert, so dass der Vektor anschließend benutzt werden kann.

Hinweise:

- Sie können davon ausgehen, dass die Methode **newWithDimension** nur mit $n \geq 0$ aufgerufen wird.
- b) Erweitern Sie die Klasse **Vector** um eine Methode **double scalarproduct(Vector q)**, welche ein Argument vom Typ **Vector** erhält und das Skalarprodukt zwischen dem aktuellen und dem übergebenen Vektor als **double** Wert zurückliefert. D.h. für zwei Objekte **p** und **q** vom Typ **Vector** berechnet der Aufruf **p.scalarproduct(q)** das Skalarprodukt zwischen **p** und **q**. Das Skalarprodukt s zweier Vektoren (x_1, \dots, x_n) und (y_1, \dots, y_n) wird gemäß der folgenden Formel berechnet:

$$s = \sum_{i=1}^n (x_i \cdot y_i)$$

Hinweise:

- Sie können davon ausgehen, dass die Methode **scalarproduct** nur mit Vektoren aufgerufen wird, welche durch die Methode **newWithDimension** erzeugt worden sind.
 - Es soll 0.0 zurückgegeben werden, falls sich die Dimension der beiden Vektoren unterscheidet. (Anmerkung: Das ist keine gute Idee für echten Code, da es so zu "stillen Fehlern" kommt, deren Ursache nur schwer zu finden ist.)
- c) Erweitern Sie die Klasse **Vector** um eine Methode **void readComponentsFromUserInput()**, welche den Benutzer für jede Komponente des aktuellen Vektors zur Eingabe eines neuen Werts auffordert, diesen einliest und das Array im Attribut des aktuellen Vektors mit diesen Werten belegt.

Hinweise:

- Sie können davon ausgehen, dass die Methode **readComponentsFromUserInput** nur mit Vektoren aufgerufen wird, welche durch die Methode **newWithDimension** erzeugt worden sind.
 - Um einen **double** Wert einzulesen, nutzen Sie die Methode **double getDouble(String prompt)** aus der Klasse **SimpleIO**, die Sie im RWTHmoodle-Lernraum "Programmierung (Übung - Tutorium) (UE) [12.04103]" finden.
- d) Schreiben Sie eine Methode **main**, welche zur Eingabe zweier Vektoren auffordert. Dazu wird zuerst die Dimension der Vektoren abgefragt. Anschließend werden beide **Vector** Objekte mit der Methode **newWithDimension** erzeugt und mit der Methode **readComponentsFromUserInput** eingelesen. Dann soll das Programm das Skalarprodukt der beiden **Vector** Objekte mit der Methode **scalarproduct** berechnen und anschließend ausgeben.

Beispiel:

Geben Sie die Dimension der Vektoren ein:

3

Geben Sie die Komponenten des ersten Vektors ein:

Geben Sie die 1-te Komponente ein:

2

Geben Sie die 2-te Komponente ein:

3

Geben Sie die 3-te Komponente ein:

4

Geben Sie die Komponenten des zweiten Vektors ein:

Geben Sie die 1-te Komponente ein:

0.5

Geben Sie die 2-te Komponente ein:

1

Geben Sie die 3-te Komponente ein:

2.5

Das Skalarprodukt der beiden Vektoren ist: 14.0

Hinweise:

- Um einen `int` Wert einzulesen, nutzen Sie die Methode `int getInt(String prompt)` aus der Klasse `SimpleIO`.
- Nutzen Sie zur Ausgabe die Methode `void output(String content)` aus der Klasse `SimpleIO`.

Aufgabe 5 (Deck 3):

(Codescape)

Lösen Sie die Räume von Deck 3 des Spiels Codescape.

Ihre Lösung für Räume dieses Codescape Decks wird nur dann für die Zulassung gezählt, wenn Sie die Lösung bis Montag, den 18.11.2019, um 12:00 Uhr abschicken.