Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Programmierung 1

Wintersemester 2023/24
Aufgabenblatt 5

17. Aufgabe (8 Punkte, Alternativen ergänzen, Nutzungsdialog)

Zur Veranschaulichung des Ergebnisses der abschließenden Teilaufgabe kann von der Veranstaltungsseite das Programm geoobjektMitIf.exe aus einem Zip-File geladen und in Windows 10 ausgeführt werden, das die zu erreichende Funktionalität veranschaulichen soll. Eventuell muss die Ausführung erlaubt werden, im Dialog „Weitere Informationen“ und dann „Trotzdem ausführen“ klicken. Da das Programm nicht in Java geschrieben ist, sieht das Interaktionsbrett etwas anders aus. Das Programm wird mit einem Klick auf „X“ rechts-oben im Interaktionsbrett beendet.

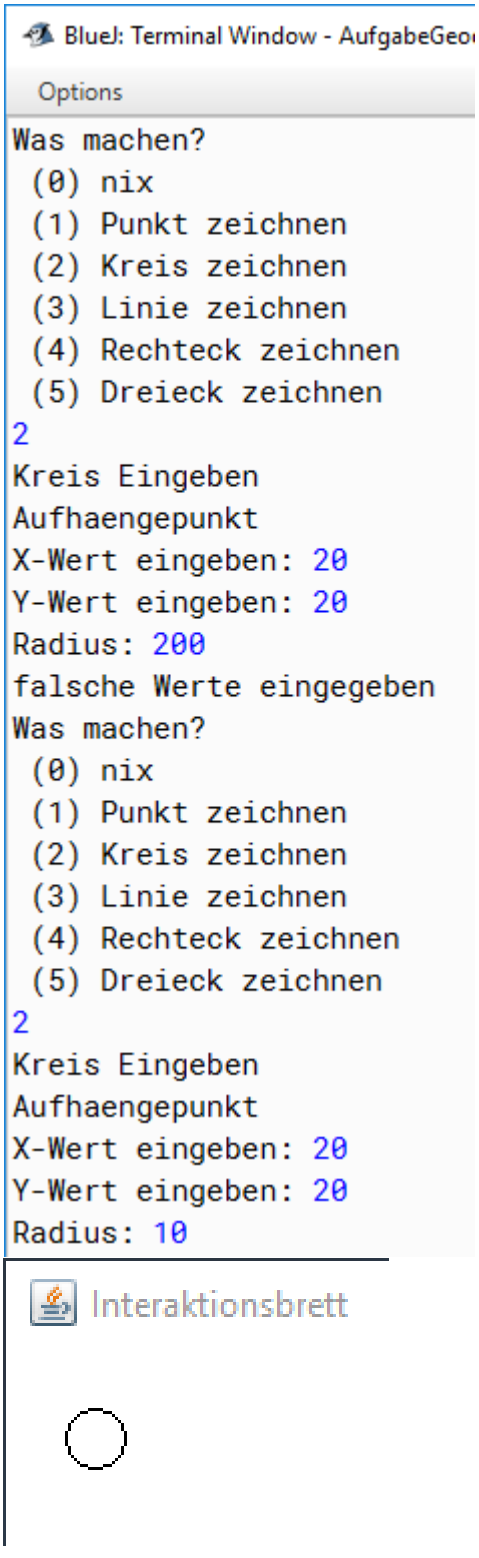
Gegeben sei ihr vorheriges Programm zur Eingabe von graphischen Objekten (Punkt, Kreis, Linie, Rechteck, Dreieck). Speichern Sie dieses Projekt als neues Projekt, das Sie jetzt bearbeiten.

a) Ergänzen Sie in jeder der folgenden Klassen eine Methode boolean istOk() die genau dann true als Ergebnis liefert, wenn jeweils die folgenden Eigenschaften gelten:
   - ein Punkt ist ok, wenn alle seine Ordinaten im sichtbaren Bereich sind, dazu soll der x-Wert positiv und kleiner 360 sowie der y-Wert positiv und kleiner als 440 sein
   - ein Kreis ist ok, wenn seine Ränder sichtbar sind; die genannten Zahlenwerte 360 und 440 sollen dabei nicht im Code der Klasse Kreis stehen, überlegen Sie sich z. B. neue Punktobjekte zu erzeugen und deren istOK-Methode zu nutzen.
   - eine Linie ist ok, wenn Start- und Endpunkt ok sind und beide Punkte nicht die gleichen x- und y-Werte haben (warum sollten Sie hier auf die Idee kommen, eine boolean equals(Punkt other)-Methode in der Klasse Punkt zu programmieren?)
   - ein Rechteck ist ok, wenn seine vier Eckpunkte ok sind
   - ein Dreieck ist ok, wenn seine drei Eckpunkte ok sind und keine der drei Punkte untereinander übereinstimmen (reicht mathematisch nicht aus, dies wird ignoriert)

b) Ändern Sie alle XYEingeben()-Methoden in der Klasse GeoobjektEingabe so ab, dass vor der Rückgabe des erzeugten Objekts zunächst geprüft wird, ob das gerade eingegebene Objekt ok ist. Ist dies nicht der Fall, wird eine Information auf der Konsole ausgegeben und statt des Objektes `null` zurückgegeben. Die Ausgabe enthält vereinfachend keine Information, welches Problem genau vorliegt, Ergonomie kommt später.

c) In der Klasse GeoobjektAusgabe könnten jetzt Probleme mit Null-Pointern auftreten (warum?), korrigieren Sie diese.

d) Ergänzen Sie in der Klasse GeoobjektAusgabe eine Methode void dialog(), mit der der Nutzer einmalig auswählen kann, welches Objekt er zeichnen möchte. Auf der rechten Seite ist ein Beispieldialog angegeben, dabei wurde dialog() zweimal aufgerufen, Nutzereingaben in blau, die Ausgabe steht getrennt weiter unten.



Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

18. Aufgabe (3 Punkte, Methoden rufen Methoden auf, Klausurähnlichkeit)

Gegeben seien folgende Klassen, notieren Sie die Ausgaben der Methoden a(), b(), c() und d() der Klasse Analyse, wenn die Methoden für ein neues Analyse-Objekt aufgerufen werden.

```
class Analyse {
    void a(){
        C c = new C();
        c.methodeC2();
    }

    void b(){
        B b = new B();
        b.methodeB3(b);
    }

    void c(){
        B b = new B();
        C ca = new C();
        b.methodeB2(ca, ca).methodeB1().methodeB3(b);
    }

    void d(){
        B b = new B();
        C ca = new C();
        b.methodeB2(new C(), new C()).methodeB2(new C(), new C());
    }
}
```

```
class C {
    EinUndAusgabe io
            = new EinUndAusgabe();

    C(){
        this.methodeC1();
        this.io.ausgeben("KC\n");
    }

    void methodeC1(){
        this.io.ausgeben("C1\n");
    }

    void methodeC2(){
        this.methodeC1();
        this.io.ausgeben("C2\n");
        this.methodeC1();
    }
}
```

```
class B {
    EinUndAusgabe io
            = new EinUndAusgabe();

    B(){
        this.methodeB1();
        this.io.ausgeben("KB\n");
    }

    B methodeB1(){
        this.io.ausgeben("B1\n");
        return this;
    }

    B methodeB2(C c1, C c2){
        c1.methodeC1();
        this.io.ausgeben("B2\n");
        c2.methodeC2();
        return new B();
    }

    void methodeB3(B b){
        this.methodeB1();
        this.io.ausgeben("B31\n");
        b.methodeB1().methodeB1();
        this.methodeB1();
        this.io.ausgeben("B32\n");
    }
}
```

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

19. Aufgabe (5 Punkte, if, Klausurähnlichkeit)

```
class IfAnalyseKlasse {

  int x = 42;

  int methode0(boolean a, boolean b){
    if(a == true && b == true){
      this.x = 0;
    }
    return this.x;
  }

  int methode1(boolean a, boolean b){
    if(a == true){
      this.x = 0;
      if(b == true){
        this.x = 0;
      }
    }
    return this.x;
  }

  int methode2(boolean a, boolean b){
    if(a == true){
      if(b == true){
        this.x = 0;
      }
    }
    return this.x;
  }

  int methode3(boolean a, boolean b){
    if(a == true){
      this.x = 0;
    } else {
      if(b == true){
        this.x = 0;
      }
    }
    return this.x;
  }

  int methode4(boolean a, boolean b){
    if(a == true){
      this.x = 0;
    }
    if(b == true){
      this.x = 0;
    }
    return this.x;
  }

  int methode5(boolean a, boolean b){
    if(a == true){
      if(b == true){
        this.x = 0;
      }
    } else {
      this.x = 0;
    }
    return this.x;
  }

  int methode6(boolean a, boolean b){
    if(a == true){
      this.x = 0;
    }
    if(b == true){
      this.x = 0;
    } else {
      this.x = 0;
    }
    return this.x;
  }

  int methode7(boolean a, boolean b){
    if(a == true && b == true){
      this.x = 0;
    } else {
    }
    return this.x;
  }

  int methode8(boolean a, boolean b){
    if(a == true){
      if(!(b == true)){
      } else {
        this.x = 0;
      }
    }
    return this.x;
  }
}
```

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Programmierung 1

Wintersemester 2023/24
Aufgabenblatt 5

Gegeben sei die Klasse IfAnalyse mit neun Methoden. Berechnen Sie selbst, welche Rückgaben die Methoden liefern, wenn Sie alle vier möglichen Varianten von Parametern *jeweils* für neue Objekte durchprobieren. Von welchen der Methoden würden Sie sagen, dass sie äquivalent sind, also bezüglich der gleichen Übergabeparameter immer das gleiche Ergebnis liefern? (m0 steht verkürzt für methode0)

a	b	m0	m1	m2	m3	m4	m5	m6	m7	m8
`true`	`true`									
`true`	`false`									
`false`	`true`									
`false`	`false`									

Randnotiz: Hier wurde für Erstsemester die wohl leichter lesbarere Form `a == true` genutzt, machen Sie sich klar, dass hier vereinfacht auch nur `a` stehen kann. Dieser Ansatz wird auch in der folgenden Aufgabe genutzt.

20. Aufgabe (2 Punkte, Ausgabe Boolescher Ausdrücke)

In dem von der Webseite zur Aufgabe erhältlichen Projekt AufgabeBoolescheAusdrueckeAuswerten befindet sich die nachfolgende Klasse BoolescherAusdruck, in der sich eine Methode befindet, um einen Booleschen Ausdruck auszuwerten.

```
class BoolescherAusdruck {
    boolean auswerten(boolean a, boolean b, boolean c) {
        return a && b || c;
        // return !(a || b || c);
    }
}
```

Schreiben Sie eine Klasse, die das Interaktionsbrett und ein Objekt der Klasse BoolescherAusdruck nutzt, um die rechts angegebene Wahrheitstafel auszugeben. Sie müssen dazu die Tabelle zeichnen und für das Ergebnis die Methode auswerten() des Booleschen Ausdrucks aufrufen.

Interaktionsbrett

a	b	c	Ergebnis
true	true	true	true
true	true	false	true
true	false	true	true
true	false	false	false
false	true	true	true
false	true	false	false
false	false	true	true
false	false	false	false

Die Nutzungsidee ist dann, dass ein beliebiger Booleschen Ausdruck mit drei Variablen in auswerten() in der return-Zeile geschrieben werden kann. Wird dann das Projekt kompiliert, ein Objekt Ihrer Klasse erzeugt und dann eine Methode aufruft, wird die zum Booleschen Ausdruck gehörige Wahrheitstafel ausgegeben.
Im obigen Beispiel könnte z. B. die obere return-Zeile auskommentiert und die Kommentarstriche in der Zeile darunter entfernt werden, um dann die Wahrheitstafel dieses Booleschen Ausdruck ausgeben zu können.
Erinnerung: Man kann beliebige Variablen x durch den folgenden Trick `String s = "" + x;` in einen String verwandeln.