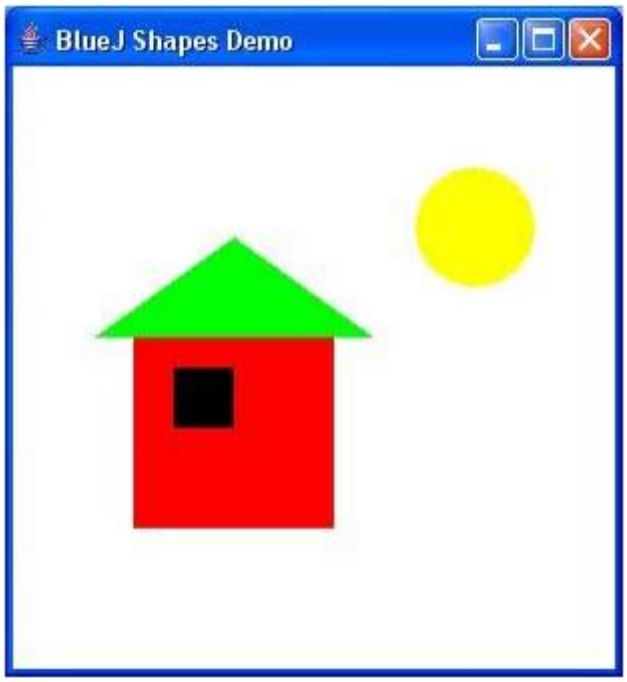
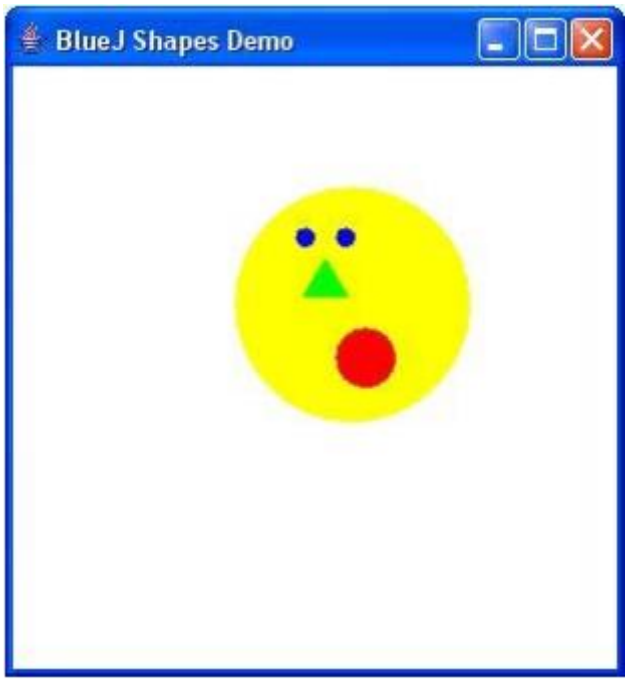
Grundlagen der IDE

(1.) Definieren Sie in der Klasse Maler zwei Methoden die nachfolgende Zeichnungen erstellen:“





(2.) Erweitern Sie die Klasse Maler um die Methode void maleHaus(int size), die ein Haus der Größe size zeichnet. Passen Sie dabei die einzelnen Elemente wie Dach, Fenster etc. entsprechend an

(3.) Erweitern Sie die Klasse Maler um die Methode void iHaveRecordedThis(), die ein Bild zeichnet. Nehmen Sie dazu vorher die Befehle mit dem Terminal auf!

(4.) Erstellen Sie ein neues Projekt TicketMachine und erstellen Sie ein neues Objekt!

(a.) Ermitteln Sie: Was bedeutet die Zahl, die Sie beim Ausführen des Konstruktors(Erstellen des Objektes) eingeben und was machen die einzelnen Methoden?

(b.)Ergänzen Sie die folgende Methode:

```
/**
 * Return the total amount of money collected by this machine.
 */
public int getTotal()
{
    return 0; // Ersetzen Sie hier diese Zeile durch Ihren eignen Text!!!
}
```

(c.)Schreiben Sie eine Methode, die bei vorgegeben(einzugebenen) Preis den aktuellen Ticketpreis neu setzt! Verwenden Sie folgende Vorlage und ergänzen Sie!

```
/**
 * Set the price of this machine's tickets to be cost (if reasonable)
 */
public void setPrice(int cost)
{
    //Fügen Sie hier den Kode hinzu!
}
```

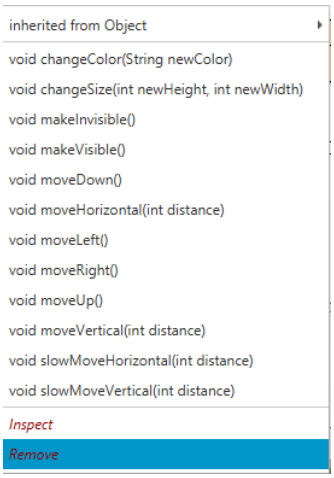
(d.) Modifizieren Sie insertMoney so, dass nur sinnvolle Beträge eingegeben werden können und testen Sie ausführlich!

(e.) Modifizieren Sie printTicket so, dass

- *ein Ticket nur dann ausgegeben wird, wenn genug Geld vorhanden ist*
- *ansonsten eine Fehlermeldung erscheint*
- *vom der Restbetrag nur der Ticketbetrag abgezogen wird anstatt diesen auf 0 zu setzen*

Testen Sie ausführlich!

(5.) (Schwer)Laden Sie das Projekt „figures“ und erweitern Sie die vorhandenen Methoden, so dass als Resultat folgende Methoden zur Verfügung stehen:



(6.) (a.)Ändern Sie in der Klasse Picture, dass jedes Objekt von Bild eine blaue statt eine gelbe Sonne bekommt, indem Sie den Quelltext der Klasse ändern!

(b.) Fügen Sie in der Klasse Picture die Zeile

private Circle sun2;

hinzu und kodieren Sie, dass eine zweite Sonne erscheint!

(c.) (Schwer) Fügen Sie der Orginalversion eine Methode sunSet() dazu, die einen Sonnenuntergang durchführt. Sie dürfen dazu die Methode slowMoveVertical verwenden!

(d.) (Schwer) Kodieren Sie, dass nach dem Sonnenuntergang eine Person zum Haus geht!

(7.)(a.) „Erstellen“ Sie 3 Studenten:
Snow White, student ID: A00234, credits: 24

Lisa Simpson, student ID: C22044, credits: 56

Charlie Brown, student ID: A12003, credits: 6

(b.) Erstellen Sie ein LabClass Objekt und fügen Sie mit enrollStudent die 3(d.) Studenten hinzu, indem Sie ihre Objektnamen angeben

(c.) Lassen Sie mit der printList-Methode des LabClass-Objektes alle Studenten ausgeben!

(d.) Machen Sie sich mit Hilfe des Inspektors mit den Eigenschaften der LabClass-Klasse vertraut!

(e.)Setzen Sie für ihr LabClass-Objekt die Eigenschaften instructor, room, time und lassen Sie alle Informationen ausgeben!

(8.)

(a.) Fügen Sie die Methoden printAuthor und printTitle zur Klasse Book hinzu. Diese sollen Autor und Titel in dem Terminal Fenster ausgeben!

(b.) Fügen Sie ein Feld(Attribut) pages vom Datentyp int hinzu, welches die Anzahl der Seiten spichert. Der Anfangswert soll im Konstruktor zusammen mit Author und Title angegeben werden.

Erstellen Sie ferner eine Methode getPages, die den aktuellen Wert von pages zurückgibt!

(c.) Erstellen Sie eine Methode printDetails, die alle Information zu dem Buch ausgibt z.B.: Title: Robinson Crusoe, Author: Daniel Defoe, Pages: 232

(d.) Erstellen Sie ein weiteres Attribut refNumber, welches vom Datentyp String ist und z.B. die Inventarnummer in einer Bücherei abspeichern kann. Diese wird im Konstruktor ohne Abfrage auf „“ gesetzt.

Zum Ändern kodieren Sie die folgende Methode:

```
public void setRefNumber(String ref){
   //Ihr Kode
}
```

Ferner kodieren Sie die entsprechende Methode getRefNumber, um den Wert des Atributes refNumber auszulesen!

(e.) Modifizieren Sie die Methode printDetails, so dass auch die refNumber ausgegeben wird. Sollte keine refNumber festgelegt worden sein, dann geben Sie „unknown“ aus!

Hinweis: Überprüfen Sie mit der Eigenschaft lenght() der Zeichenkette die Länge von refNumber!

(f.) Ändern Sie die Methode setRefNumber, so dass diese nur Eingaben akzeptiert, die mindestens 3 Zeichen lang sind. Im anderen Fall soll eine Fehleraufgabe erfolgen und refNumber unverändert bleiben.

(g.) Fügen Sie eine weitere Eigenschaft borrowed mit Datentyp Integer hinzu, welches die Anzahl speichert, wie oft das Buch ausgeliehen worden ist.

Kodieren Sie eine Methode borrow(), welche den Wert von borrowed erhöht, wenn eine weitere Buchausleihe erfolgte.

Kodieren Sie getBorrowed(), um auszulesen, wie oft das Buch ausgeliehen worden ist(der Wert von borrowed wird ausgelesen)!

Ändern sie die Methode printDetails, so dass auch der Wert von borrowed ausgegeben wird!

(h.) Fügen Sie ein Attribut courseText mit Datentyp boolean hinzu, welches speichert, ob ein Buch als Textbuch in dem Kurs verwendet wird. Setzen Sie diesen Wert im Konstruktor!

Kodieren Sie ferner eine Methode isCourseText für den Zugriff auf die Variable courseText!

(9.) (a.) Erstellen Sie ein neues Projet HeaterExcerise und erstellen dort die Klasse Heater mit Attributen temperature mit Datentyp double! Definieren Sie einen Konstruktor ohne Parameter und setzen Sie die Temperatur auf 15 dort.

Definieren Sie die Methoden warmer und cooler, die die Temperatur um jeweils 5 Grad erhöht.

Definieren Sie eine Zugriffsmethode auf das Attribute temperature, um auf dieses zugreifen zu können!

(b.) Fügen Sie der Heater-Klasse drei neue Attribute mit Datentyp double hinzu:
min, max und increment.

Die Werte von min und max werden von dem Konstruktor gesetzt, während increment dort immer auf 5 gesetzt wird.

Ändern Sie die Methoden von warmer() und colder(), so dass diese den Wert von increment zum Verändern der Temperatur verwenden!

Stellen Sie sicher, dass in beiden Methoden der neue Wert zwischen max und min liegt! Ansonsten bleibt die Temperatur gleich und es wird eine Fehlermeldung ausgegeben!

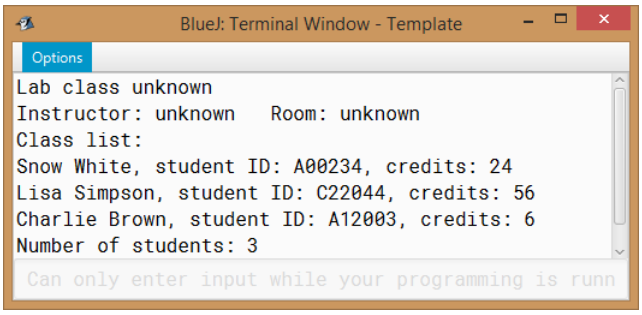
Fügen Sie eine weitere Methode setIncrement hinzu, um den Wert von increment zu setzen. Stellen Sie sicher, dass hier nur numerische Werte größer als Null eingegeben werden!

Testen Sie das Programm mindestens 15 Minuten mit verschiedenen Fällen!

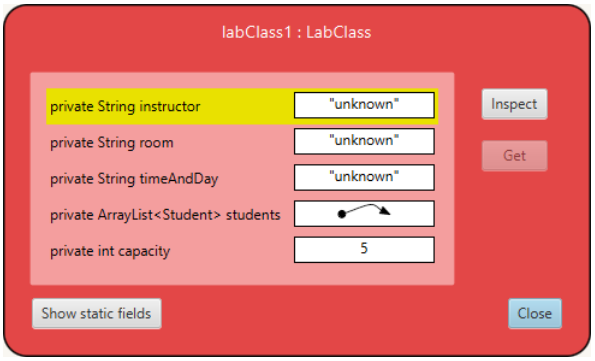
Grundlagen der IDE - Lösungen

(7.)(a.)

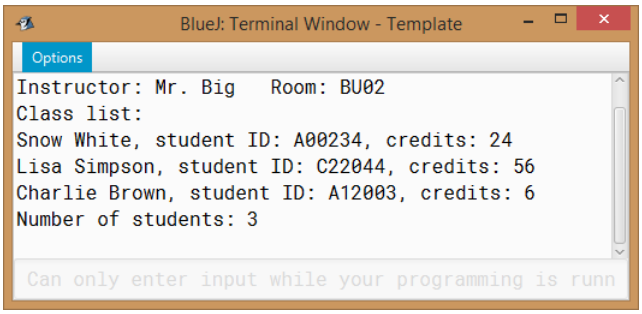
(c.)



(d.)



(e.)


BlueJ: Terminal Window - Template
Options
Instructor: Mr. Big Room: BU02
Class list:
Snow White, student ID: A00234, credits: 24
Lisa Simpson, student ID: C22044, credits: 56
Charlie Brown, student ID: A12003, credits: 6
Number of students: 3
Can only enter input while your programming is runn