Grundlagen der Informatik 4 (GIT IV) Übung

Bearbeited von:

Name	Matrikelnummer	Nummer des Übungsblatts
Maximilian Bradstädter	1007486	6

Aufgabe 1:

```
public class Zahlungsabwicklung {
 protected boolean paypal = false; /* :) */
 protected double Betrag;

 private String clean_string(String s) {
  return s.trim().toLowerCase();
 }

 public Zahlungsabwicklung(final String _Zahlungsart, final double _Betrag) {
  if (clean_string(_Zahlungsart) == "paypal") {
   // if (clean_string(_Zahlungsart).compareTo("paypal") == 0) {
   this.paypal = true;
  }
  this.Betrag = _Betrag;
 }

 protected double calc_gebuehren() {
  if (paypal) {
   return (this.Betrag * 0.04) + 5;
  } else {
   if (this.Betrag < 100) {
    return this.Betrag * 0.05;
   } else {
```

```
      return (this.Betrag * 0.02) + 8;
    }
  }
 }

 public double berechneGesamtbetrag() {
  return this.Betrag + calc_gebuehren();/* why '+' ??? */
 }

 public static void main(String[] args) {
  Zahlungsabwicklung z1 = new Zahlungsabwicklung("paypal", 100);
  Zahlungsabwicklung z2 = new Zahlungsabwicklung("Kreditkarte", 100);

  System.err.println(z1.berechneGesamtbetrag());
  System.err.println(z2.berechneGesamtbetrag());

 }
}
```

Aufgabe 2:

```
public class Zahlungsabwicklung {
 protected boolean paypal = false; /* :) */
 protected double Betrag;

 private String clean_string(String s) {
  return s.trim().toLowerCase();
 }

 public Zahlungsabwicklung(final String _Zahlungsart, final double _Betrag) {
```

```
  if (clean_string(_Zahlungsart) == "paypal") {
    // if (clean_string(_Zahlungsart).compareTo("paypal") == 0) {
    this.paypal = true;
  }
  this.Betrag = _Betrag;
}

protected double calc_gebuehren() {
  if (paypal) {
    return (this.Betrag * 0.04) + 5;
  } else {
    if (this.Betrag < 100) {
      return this.Betrag * 0.05;
    } else {
      return (this.Betrag * 0.02) + 8;
    }
  }
}

public double berechneGesamtbetrag() {
  return this.Betrag + calc_gebuehren();/* why '+' bei betrag ??? */
}

public static void erklaereGebuehren(final String _Zahlungsart, final double _Betrag) {
  final String erklaerung = """
      public Zahlungsabwicklung(final String _Zahlungsart, final double _Betrag) {
        if (clean_string(_Zahlungsart) == "paypal") {
          this.paypal = true;
```

```
      }
      this.Betrag = _Betrag;
    }

    protected double calc_gebuehren() {
      if (paypal) {
        return (this.Betrag * 0.04) + 5;
      } else {
        if (this.Betrag < 100) {
          return this.Betrag * 0.05;
        } else {
          return (this.Betrag * 0.02) + 8;
        }
      }
    }

    public double berechneGesamtbetrag() {
      return this.Betrag + calc_gebuehren();/* why '+' ??? */
    }
      """;

  System.out.println(erklaerung);
  System.out.format("\nIn deinem fall würde es zu %f euro evaluieren\n",
    new Zahlungsabwicklung(_Zahlungsart, _Betrag).berechneGesamtbetrag());
}

public static void main(String[] args) {
  erklaereGebuehren("Paypal", 100.001);
```

```
  erklaereGebuehren("Kreditkarte", 100.001);
 }
}
```

Aufgabe 3:

Code:

```
public class Zahlungsabwicklung {
 protected Zahlungsart Zahlart = Zahlungsart.UNBEKANNT; /* :) */
 protected double Betrag;

 enum Zahlungsart {
  PAYPAL,
  KREDITKARTE,
  BANKUEBERWEISUNG,
  UNBEKANNT;

  static Zahlungsart fromString(String s) {
   return switch (s) {
    case "paypal" -> PAYPAL;
    case "kreditkarte" -> KREDITKARTE;
    case "bankueberweisung" -> BANKUEBERWEISUNG;
    default -> UNBEKANNT;
   };
  }
 }

 private String clean_string(String s) {
  return s.trim().toLowerCase();
```

```
}

public Zahlungsabwicklung(final String _Zahlungsart, final double _Betrag) {
 this.Zahlart = Zahlungsart.fromString(clean_string(_Zahlungsart));
 this.Betrag = _Betrag;
}

protected double calc_gebuehren() {
 switch (Zahlart) {
  case PAYPAL:
   return (this.Betrag * 0.04) + 5;
  case KREDITKARTE:
   if (this.Betrag < 100) {
    return this.Betrag * 0.05;
   } else {
    return (this.Betrag * 0.02) + 8;
   }
  default:
   return this.Betrag;
 }
}

public double berechneGesamtbetrag() {
 return this.Betrag + calc_gebuehren();/* why '+' bei betrag ??? */
}

public static void erklaereGebuehren(final String _Zahlungsart, final double _Betrag) {
 final String erklaerung = """
```

```
protected Zahlungsart Zahlart = Zahlungsart.UNBEKANNT; /* :) */
protected double Betrag;

enum Zahlungsart {
 PAYPAL,
 KREDITKARTE,
 BANKUEBERWEISUNG,
 UNBEKANNT;

 static Zahlungsart fromString(String s) {
  return switch (s) {
   case "paypal" -> PAYPAL;
   case "kreditkarte" -> KREDITKARTE;
   case "bankueberweisung" -> BANKUEBERWEISUNG;
   default -> UNBEKANNT;
  };
 }
}

private String clean_string(String s) {
 return s.trim().toLowerCase();
}

public Zahlungsabwicklung(final String _Zahlungsart, final double _Betrag) {
 this.Zahlart = Zahlungsart.fromString(clean_string(_Zahlungsart));
 this.Betrag = _Betrag;
}
```

```
    protected double calc_gebuehren() {
     switch (Zahlart) {
      case PAYPAL:
       return (this.Betrag * 0.04) + 5;
      case KREDITKARTE:
       if (this.Betrag < 100) {
        return this.Betrag * 0.05;
       } else {
        return (this.Betrag * 0.02) + 8;
       }
      default:
       return this.Betrag;
     }
    }

    public double berechneGesamtbetrag() {
     return this.Betrag + calc_gebuehren();/* why '+' bei betrag ??? */
    }
         """;

  System.out.println(erklaerung);
  System.out.format("\nIn deinem fall würde es zu %f euro evaluieren\n",
    new Zahlungsabwicklung(_Zahlungsart, _Betrag).berechneGesamtbetrag());
}

public static void main(String[] args) {
  /* nix gefragt */
}
```

}

Zu der aktualisierung bzw Erweiterbarkeit des momentanen codes fällt mir nix besonderes auf, da dieser relativ gut wart/erweiterbar ist, sofern es im Rahmen solcher aufgaben bleibt.

Aufgabe 4:

Durch vererbung und Polymorphism ist es nun einfacher im großen stil neue Zahlungsarten hinzuzufügen. Dies spart zeit. Neue schritte die benötigt werden um eine Zahlungsart hinzuzufügen:

- Im Zahlungsarten ordner eine neue java datei hinzufügen
- Hier eine klasse deklarieren die „Zahlungsart“ als Oberklasse hat.
- Hier die abstrakten funktionen implementieren

Code:

ZahlungsabwicklungOOP.java:

```
import java.util.ArrayList;
import Zahlungsarten.*;

public class ZahlungsabwicklungOOP {
  public static void main(String[] args) {
    ArrayList<Zahlungsart> Zahlarten = new ArrayList<Zahlungsart>();
    Zahlarten.add(new Paypal());
    Zahlarten.add(new Kreditkarte());
    Zahlarten.add(new Bankueberweisung());

    for (Zahlungsart Zahlart : Zahlarten) {
      Zahlart.erklaereGebuehren(100); /* prints berechneGesamtbetrag result, so unnecessary here */
    }
  }
```

```
}
```

Zahlungsarten\Bankueberweisung.java:

```
package Zahlungsarten;

public class Bankueberweisung extends Zahlungsart {
  public void erklaereGebuehren(final double _Betrag) {
    final String erklaerung = """
       public double berechneGesamtbetrag(final double _Betrag) {
         return _Betrag;
       }
       """;

    System.out.println(erklaerung);
    System.out.format("In deinem fall würde es zu %f euro evaluieren\n\n" + //
        "",
       new Bankueberweisung().berechneGesamtbetrag(_Betrag));
  }

  public double berechneGesamtbetrag(final double _Betrag) {
    return _Betrag;
  }
}
```

Zahlungsarten\Kreditkarte.java:

```
package Zahlungsarten;
```

```
public class Kreditkarte extends Zahlungsart {
 public void erklaereGebuehren(final double _Betrag) {
  final String erklaerung = """
    public double berechneGesamtbetrag(final double _Betrag) {
     if (_Betrag < 100) {
      return _Betrag + _Betrag * 0.05;
     } else {
      return _Betrag + (_Betrag * 0.02) + 8;
     }
    }
          """;

  System.out.println(erklaerung);
  System.out.format("In deinem fall würde es zu %f euro evaluieren\n\n" + //
    "",
    new Kreditkarte().berechneGesamtbetrag(_Betrag));
 }

 public double berechneGesamtbetrag(final double _Betrag) {
  if (_Betrag < 100) {
   return _Betrag + _Betrag * 0.05;
  } else {
   return _Betrag + (_Betrag * 0.02) + 8;
  }
 }
}
```

Zahlungsarten\Paypal.java:

```
package Zahlungsarten;

public class Paypal extends Zahlungsart {
 public void erklaereGebuehren(final double _Betrag) {
  final String erklaerung = """
     public double berechneGesamtbetrag(final double _Betrag) {
      return _Betrag + (_Betrag * 0.04) + 5;
     }
         """;

  System.out.println(erklaerung);
  System.out.format("In deinem fall würde es zu %f euro evaluieren\n\n",
     new Paypal().berechneGesamtbetrag(_Betrag));
 }

 public double berechneGesamtbetrag(final double _Betrag) {
  return _Betrag + (_Betrag * 0.04) + 5;
 }
}
```

Zahlungsarten\Zahlungsart.java:

```
package Zahlungsarten;

public abstract class Zahlungsart {
```

```
  /*
   * ungewünscht?
   * protected double Betrag;
   *
   * public Zahlungsart(double _Betrag) {
   * this.Betrag = _Betrag;
   * }
   */

  public abstract void erklaereGebuehren(final double _Betrag);

  public abstract double berechneGesamtbetrag(final double _Betrag);
}
```