Übungsblatt:	6	**1. Abgabepartner:**	Matthias Wolff	(458 766)
		2. Abgabepartner:	Anton Mende	(461 328)
		3. Abgabepartner:	Anika Herbermann	(461 655)

Aufgabe 18

Relation "Kunde"

kundenId	name
1	Aaron Eckhart
2	Michael Caine
3	Maggie Gyllenhaal

Relation "Pizza"

pizzaId	name	preis
1	Hollandaise	5,80
2	Cipolla	5,80
3	Tonno	6,80
4	Zingara	6,80
5	Gyros	6,80
6	Toscana	7,50

Relation "Bestellung"

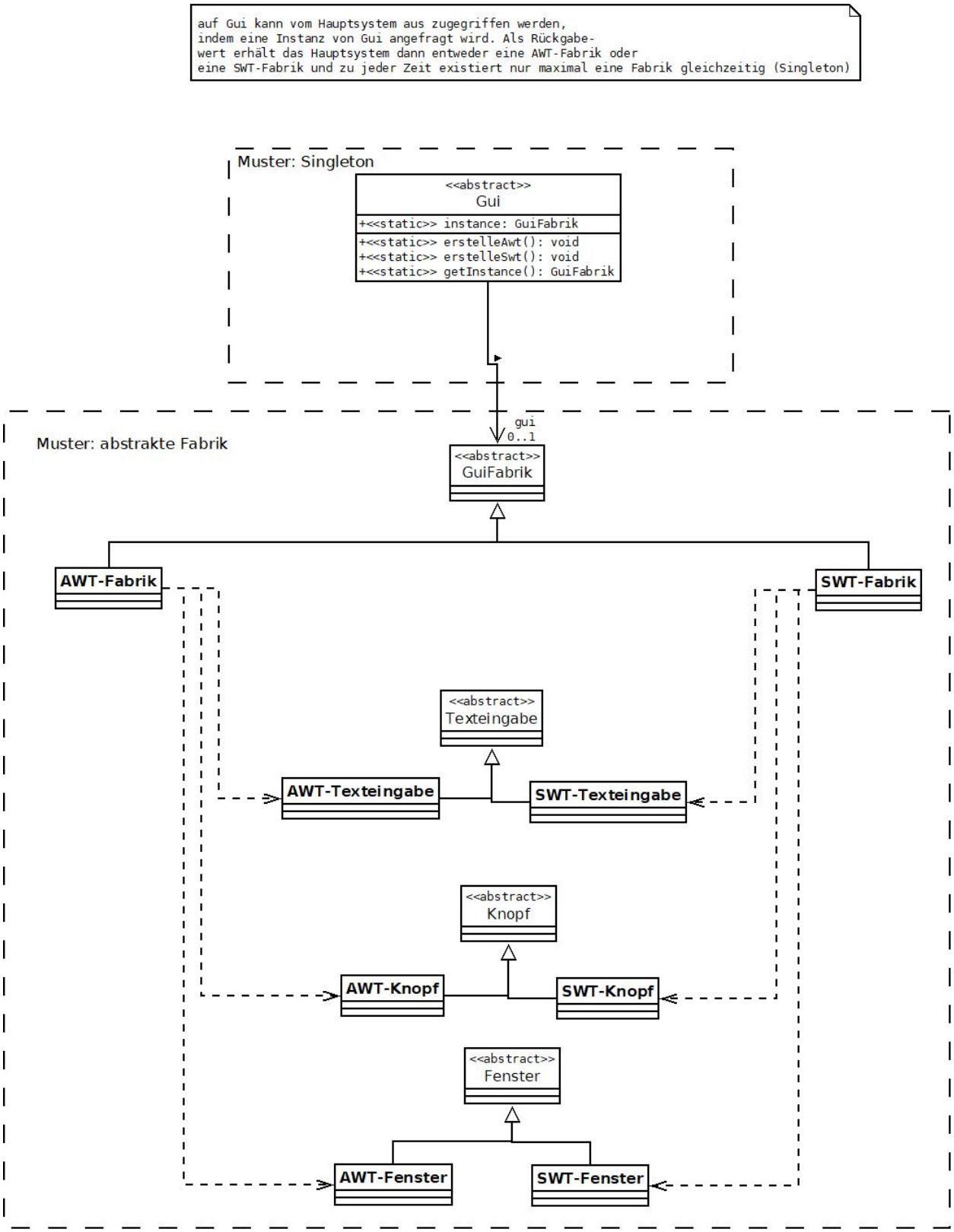
buchungsId	buchungsZeitpunkt	pizzaId	kundenId
1	17.06.2017 11:58:29	2	1
2	17.06.2017 14:28:58	5	1
3	17.06.2017 16:19:22	6	2
4	17.06.2017 18:21:31	1	3

```java
import java.util.Collection;
import java.util.ArrayList;

class Kunde {
    protected String kundenId;
    protected String name;
    protected Collection<Bestellung> buchung = new ArrayList<Bestellung>();

    public Kunde(String kId, String n){
        kundenId = kId;
        name = n;
    }

    public String getKundenId(){return kundenId;}
    public void setKundenId(String kId){kundenId = kId;}

    public String getName(){return name;}
    public void setName(String newName) {name = newName;}

    public Collection<Bestellung> getBuchung() {
        return buchung;
    }
    public void setBuchung(Collection<Bestellung> coll) {
        buchung = coll;
    }
    public void addBuchung(Bestellung vorgang) {
        buchung.add(vorgang);
    }
    public void removeBuchung(Bestellung vorgang) {
        buchung.remove(vorgang);
    }
}
```

```java
import java.util.ArrayList;
import java.util.Collection;

class Pizza {
    protected int pizzaId;
    protected double preis;
    protected String name;
    protected Collection<Bestellung> buchung = new ArrayList<Bestellung>();

    public Pizza(int pId, double p, String n){
        pizzaId = pId;
        preis = p;
        name = n;
    }

    public int getPizzaId(){return pizzaId;}
    public void setPizzaId(int pId){pizzaId=pId;}

    public double getPreis(){return preis;}
    public void setPreis(double p) {preis=p;}

    public String getName(){return name;}
    public void setName(String n){name=n;}

    public Collection<Bestellung> getBuchung() {
        return buchung;
    }
    public void setBuchung(Collection<Bestellung> coll) {
        buchung = coll;
    }
    public void addBuchung(Bestellung vorgang) {
        buchung.add(vorgang);
    }
    public void removeBuchung(Bestellung vorgang) {
        buchung.remove(vorgang);
    }
}
```

```
import java.time.LocalDateTime;
class Bestellung{
    protected int buchungsId;
    protected LocalDateTime zeitpunkt;
    protected Kunde kunde;
    protected Pizza pizza;

    Bestellung(LocalDateTime z, int bId, Kunde k, Pizza p){
        zeitpunkt = z;
        buchungsId = bId;
        kunde=k;
        pizza=p;
    }

    public int getBuchungsId(){return buchungsId;}
    public void setBuchungsId(int bId){buchungsId =bId;}

    public LocalDateTime getZeitpunkt(){return zeitpunkt;}
    public void setZeitpunkt(LocalDateTime zp){zeitpunkt = zp;}

    public Kunde getKunde(){ return kunde;}
    public void setKunde(Kunde k){kunde = k;}

    public Pizza getPizza(){return pizza;}
    public void setPizza(Pizza p){pizza =p;}
}
```

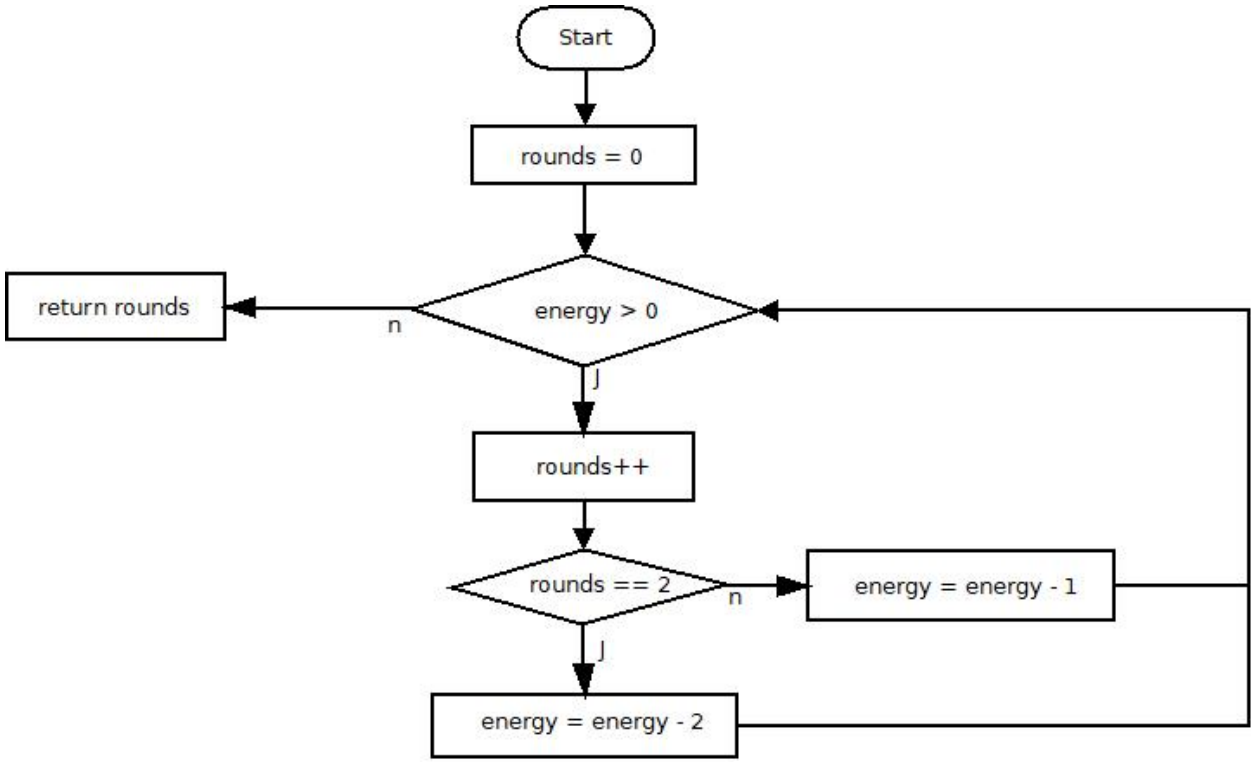
Aufgabe 19


auf Gui kann vom Hauptsystem aus zugegriffen werden,
indem eine Instanz von Gui angefragt wird. Als Rückgabe-
wert erhält das Hauptsystem dann entweder eine AWT-Fabrik oder
eine SWT-Fabrik und zu jeder Zeit existiert nur maximal eine Fabrik gleichzeitig (Singleton)
Muster: Singleton
<<abstract>>
Gui
+<<static>> instance: GuiFabrik
+<<static>> erstelleAwt(): void
+<<static>> erstelleSwt(): void
+<<static>> getInstance(): GuiFabrik
gui
0..1
Muster: abstrakte Fabrik
<<abstract>>
GuiFabrik
AWT-Fabrik
SWT-Fabrik
<<abstract>>
Texteingabe
AWT-Texteingabe
SWT-Texteingabe
<<abstract>>
Knopf
AWT-Knopf
SWT-Knopf
<<abstract>>
Fenster
AWT-Fenster
SWT-Fenster

```
1  import java.awt.*;
   import org.eclipse.swt.*;
4
   public abstract class Gui {
       private static GuiFabrik instance;
7
       public static void erstelleAwt() {
           Gui.instance = new AWTFabrik();
       }
10
       public static void erstelleSwt() {
           Gui.instance = new SWTFabrik();
13     }

       public static GuiFabrik getInstance() {
16         if(Gui.instance == null){
               //Default-Framework
               Gui.erstelleAwt();
19         }
       return Gui.instance;

22     }
   }
```

Aufgabe 20

a)



b) [rounds, rounds = 0, return rounds]
[rounds, rounds = 0, rounds++]
[rounds, rounds++, rounds == 2]
[rounds,rounds++,rounds++]
[rounds, rounds++, return rounds]
[energy, energy = energy - 1, energy > 0]
[energy, energy = energy - 1, energy = energy - 2]
[energy, energy = energy - 1, energy = energy - 1]
[energy, energy = energy - 2, energy > 0]
[energy, energy = energy - 2, energy = energy - 1]
[energy, energy = energy - 2, energy = energy - 2] (nicht erreichbar)

c) "Energy = 0" durchläuft Kette 1, "Energy = 5" durchläuft Kette 2 - 10.