FH-OÖ Hagenberg/HSD
SDP3, WS 2019
Übung 7

Name(1): Daniel Weyrer — Abgabetermin:

Name(2): Viktoria Streibl — Punkte:

Übungsgruppe: Gruppe 1 — korrigiert:

Geschätzter Aufwand in Ph: 6 | 6 — Effektiver Aufwand in Ph: 6 | 5

Beispiel 1 (24 Punkte) Kaffeeautomat: Entwerfen Sie aus der nachfolgenden Spezifikation ein Klassendiagramm, instanzieren Sie dieses und implementieren Sie die Funktionalität entsprechend. Verwenden Sie dabei das Decorator-Pattern:

Ein Kaffeeautomat bietet verschiedene Kaffeesorten (Verlängerter, Espresso, Koffeinfrei) mit entsprechenden Zutaten (Zucker, Milch u. Schlagobers) an. Die Kaffeesorten und Zutaten haben jeweils unterschiedliche Preise und eine entsprechende Beschreibung. Eine Methode `GetCost()` liefert den Gesamtpreis des ausgewählten Kaffees und die Methode `GetDescription()` liefert dazu die entsprechende Beschreibung als `std::string` um z.B. folgende Ausgaben auf `std::cout` zu ermöglichen:

```
Espresso: Zucker, Schlagobers 2.89 Euro
Verlängerter: Zucker, Milch 2.93 Euro
Koffeinfrei: Milch, Milch, Schlagobers 3.15 Euro
```

Die Beschreibung und die Preise werden in einer separaten Preisliste (Konstanten in Header, Klasse, oder Namespace) festgelegt. Zutaten können mehrfach gewählt werden!

Achten Sie beim Design darauf, dass zusätzliche Kaffeesorten und Zutaten hinzugefügt werden können, ohne die bereits bestehenden Klassen verändern zu müssen. Beweisen Sie dies durch das Hinzufügen der Kaffeesorte "Mocca" und der Zutat "Sojamilch".

Implementieren Sie einen Testtreiber der verschiedene Kaffees mit unterschiedlichen Zutaten erzeugt, alle Methoden ausreichend testet und anschließend deren Beschreibung auf `std::cout` ausgibt.

Implementieren Sie weiters eine Klasse `CoffeePreparation` die nach dem FIFO-Prinzip arbeitet und folgende Schnittstelle aufweist:

```
1 void Prepare(/*Coffee*/);         //adds and prepares a coffee
2 void Display(std::ostream& os);   //outputs all coffees in preparation
3 /*Coffee*/ Finished();               //removes the prepared coffee
```

Testen Sie die Klasse ebenfalls ausführlich im Testtreiber!

Allgemeine Hinweise: Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung!** Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit Kommentaren! Testen Sie ihre Implementierungen ausführlich! Geben Sie den **Testoutput** mit ab!

SDP - Exercise 07

winter semester 2019/20

Viktoria Streibl - S1810306013
Daniel Weyrer - S1820306044

January 6, 2020

Contents

1 Organizational

1.1 Team

- Viktoria Streibl - S1810306013
- Daniel Weyrer - S1820306044

1.2 Roles and responsibilities

1.2.1 Jointly

- Planning
- Documentation
- Systemdocumentation

1.2.2 Viktoria Streibl

- Object
- Pricelist
- Coffee Sorts
- Ingredient Sorts

1.2.3 Daniel Weyrer

- Coffeemachine
- TestDriver
- CoffeePreparation
- Ingredient

1.3 Effort

1.3.1 Viktoria Streibl

- estimated: 6 ph
- actually: 5 ph

1.3.2 Daniel Weyrer

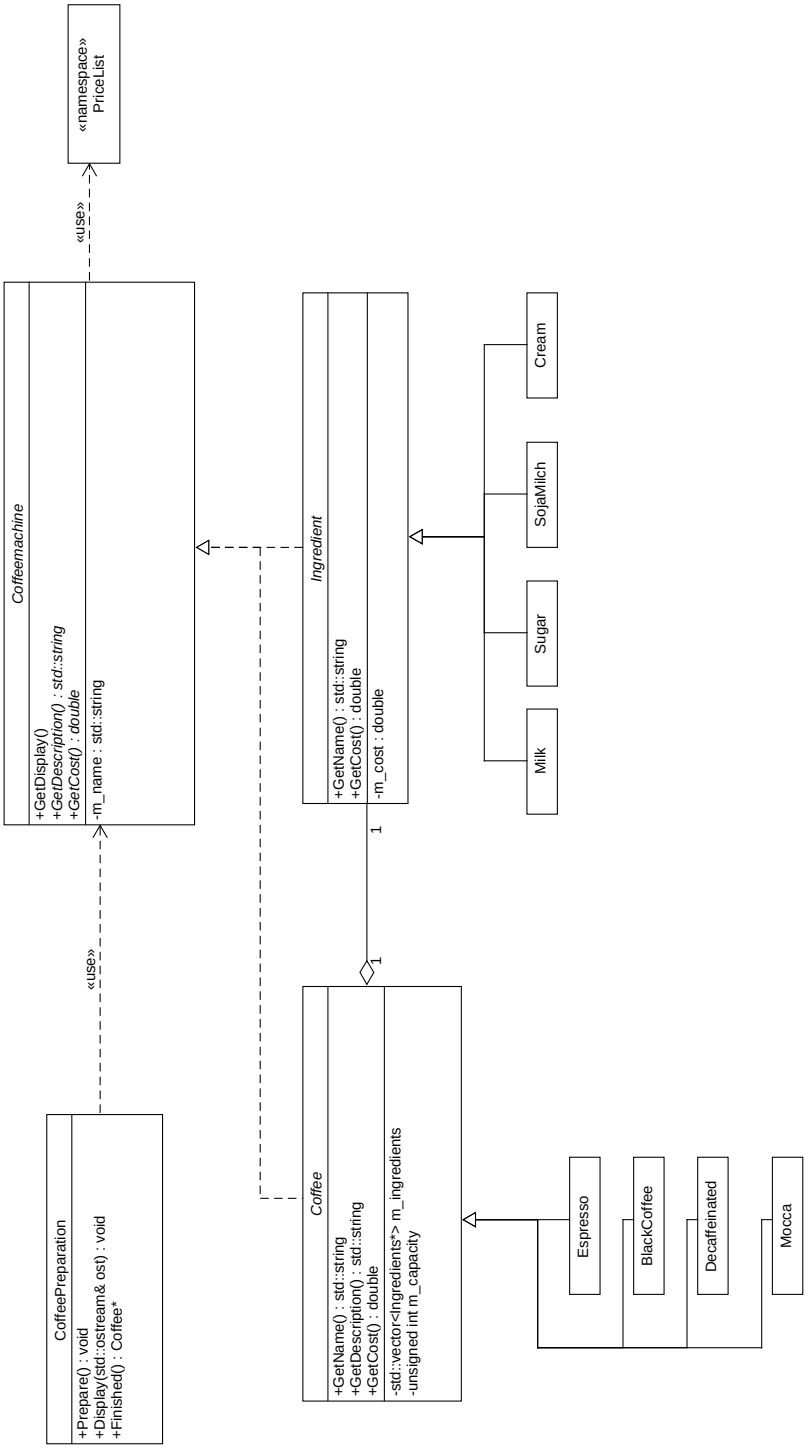
- estimated: 6 ph
- actually: 6 ph

2 Requirenment Definition(System Specification)

This Coffeemachine should work like a normal Coffeemachine. It is a simulation to order different sort of coffee and add several ingredients. Depending on the selection the price will be displayed.

3 System Design

3.1 Classdiagram



3.2 Design Decisions

3.2.1 PriceList

We decided to use an extra file to manage the different prices. This makes it easier to add change the prices afterwards.

4 Component Design

4.1 CoffeePreparation

It contains following Methods:

- Prepare
- Display
- Finished

4.2 Coffeemachine

It contains following Methods:

- GetDescription
- GetCost
- GetName

4.3 Pricelist

It contains a namespace where the prices of the different ingredients are declared.

4.4 Ingredient

4.5 Coffee Sorts

There are different kinds of coffee, the following are implemented:

- Espresso
- Black Coffee
- Decaffeinated
- Mocca

4.6 Ingredient Sorts

There are different kinds of coffee, the following are implemented:

- Milk
- Sugar
- Cream
- Soja Milk

They also have following methods:

- GetDescription
- GetCost

5 Source Code

5.1 CoffeePreparation

5.1.1 CoffeePreparation.h

```
1  /* ------------------------------------------------------------------------
2  | Workfile : CoffeePreparation.h
3  | Description : [ HEADER ]
4  | Name : Daniel Weyrer      PKZ : S1820306044
5  | Date : 06.01.20
6  | Remarks : -
7  | Revision : 0
8  | ------------------------------------------------------------------------ */
9
10
11 #ifndef COFFEEPREPARATION_H
12 #define COFFEEPREPARATION_H
13
14 #include "Object.h"
15
16 class CoffeePreparation : public Object {
17 };
18
19 #endif //!COFFEEPREPARATION_H
```

5.1.2 CoffeePreparation.cpp

```
1  /* ------------------------------------------------------------------------
2  | Workfile : CoffeePreparation.cpp
3  | Description : [ SOURCE ]
4  | Name : Daniel Weyrer      PKZ : S1820306044
5  | Date : 06.01.20
6  | Remarks : -
7  | Revision : 0
8  | ------------------------------------------------------------------------ */
9
10 #include "CoffeePreparation.h"
```

5.2 Coffeemachine

5.2.1 Coffeemachine.h

```
/* ----------------------------------------------------------------------
| Workfile : Coffeemachine.h
| Description : [ HEADER ]
| Name : Daniel Weyrer      PKZ : S1820306044
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ---------------------------------------------------------------------- */

#ifndef COFFEEMACHINE_H
#define COFFEEMACHINE_H

#include "Object.h"


class Coffeemachine : public Object{
public:
  void GetDisplay();
  virtual std::string GetDescription() = 0;
  virtual double GetCost() = 0;

private:
  std::string m_name;

};


#endif //!COFFEEMACHINE_H
```

5.2.2 Coffeemachine.cpp

```
/* ----------------------------------------------------------------------
| Workfile : Coffeemachine.cpp
| Description : [ SOURCE ]
| Name : Daniel Weyrer      PKZ : S1820306044
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ---------------------------------------------------------------------- */
#include <iostream>
#include "Coffeemachine.h"



void Coffeemachine::GetDisplay() {
}

void Coffeemachine::SetName(std::string const& name) {
  m_name = name;
}

std::string Coffeemachine::GetName() const {
  return m_name;
}
```

5.3 Espresso

5.3.1 Espresso.h

```
/* ------------------------------------------------------------------------
| Workfile : Espresso.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef ESPRESSO_H
#define ESPRESSO_H

#include <string>

#include "Pricelist.h"
#include "Coffee.h"

class Espresso : public Coffee {

  std::string GetName() override;
  std::string GetDescription() override;
  double GetCost() override;
};

#endif //!ESPRESSO_H
```

5.3.2 Espresso.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Espresso.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Espresso.h"

using namespace pricelist;

std::string Espresso::GetName() {
  return "Espresso";
}

std::string Espresso::GetDescription() {
  std::string desc = "";

  return desc;
}

double Espresso::GetCost() {
  double price = coffee::espresso;

  return price;
}
```

5.4 BlackCoffee

5.4.1 BlackCoffee.h

```
/* ------------------------------------------------------------------------
| Workfile : BlackCoffee.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef BLACKCOFFEE_H
#define BLACKCOFFEE_H

#include <string>

#include "Pricelist.h"
#include "Coffee.h"

class BlackCoffee : public Coffee {

  std::string GetName() override;
  std::string GetDescription() override;
  double GetCost() override;
};

#endif //!BLACKCOFFEE_H
```

5.4.2 BlackCoffee.cpp

```
/* ------------------------------------------------------------------------
| Workfile : BlackCoffee.h
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "BlackCoffee.h"

using namespace pricelist;

std::string BlackCoffee::GetName() {
  return "Black Coffee";
}

std::string BlackCoffee::GetDescription() {
  std::string desc = "";

  return desc;
}

double BlackCoffee::GetCost() {
  double price = coffee::blackcoffee;

  return price;
}
```

5.5 Decaffeinated

5.5.1 Decaffeinated.h

```
/* ------------------------------------------------------------------------
| Workfile : Decaffeinated.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef DECAFFEINATED_H
#define DECAFFEINATED_H

#include <string>

#include "Pricelist.h"
#include "Coffee.h"

class Decaffeinated : public Coffee {

  std::string GetName() override;
  std::string GetDescription() override;
  double GetCost() override;
};

#endif //!DECAFFEINATED_H
```

5.5.2 Decaffeinated.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Decaffeinated.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Decaffeinated.h"

using namespace pricelist;

std::string Decaffeinated::GetName() {
  return "Decaffeinated";
}

std::string Decaffeinated::GetDescription() {
  std::string desc = "";

  return desc;
}

double Decaffeinated::GetCost() {
  double price = coffee::decaffeinated;

  return price;
}
```

5.6 Mocca

5.6.1 Mocca.h

```
/* ------------------------------------------------------------------------
| Workfile : Mocca.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef COFFEE_H
#define COFFEE_H

#include <string>

#include "Pricelist.h"
#include "Coffee.h"

class Mocca : public Coffee {

  std::string GetName() override;
  std::string GetDescription() override;
  double GetCost() override;
};

#endif //!COFFEE_H
```

5.6.2 Mocca.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Mocca.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Mocca.h"

using namespace pricelist;

std::string Mocca::GetName() {
  return "Mocca";
}

std::string Mocca::GetDescription() {
  std::string desc = "";

  return desc;
}

double Mocca::GetCost() {
  double price = coffee::mocca;

  return price;
}
```

5.7 Ingredient

5.7.1 Ingredient.h

```
/* ------------------------------------------------------------------------
| Workfile : Ingredient.h
| Description : [ HEADER ]
| Name : Daniel Weyrer      PKZ : S1820306044
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef INGREDIENT_H
#define INGREDIENT_H

#include "Coffeemachine.h"
class Ingredient : public Coffeemachine {
};

#endif //!INGREDIENT_H
```

5.7.2 Ingredient.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Ingredient.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Ingredient.h"
```

5.8 Milk

5.8.1 Milk.h

```
/* ------------------------------------------------------------------------
| Workfile : Milk.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef MILK_H
#define MILK_H

#include <string>

#include "Pricelist.h"
#include "Ingredient.h"

class Milk : public Ingredient {
  std::string GetName() override;
  double GetCost() override;
};

#endif //!MILK_H
```

5.8.2 Milk.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Milk.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Milk.h"

using namespace pricelist;

std::string Milk::GetName() {
  return "Milk";
}
double Milk::GetCost() {
  return ingredients::milk;
}
```

5.9 Sugar

5.9.1 Sugar.h

```
/* ------------------------------------------------------------------------
| Workfile : Sugar.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef SUGAR_H
#define SUGAR_H

#include <string>

#include "Pricelist.h"
#include "Ingredient.h"

class Sugar : public Ingredient {
  std::string GetName() override;
  double GetCost() override;
};

#endif //!SUGAR_H
```

5.9.2 Sugar.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Sugar.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Sugar.h"

using namespace pricelist;

std::string Sugar::GetName() {
  return "Sugar";
}
double Sugar::GetCost() {
  return ingredients::sugar;
}
```

5.10 Cream

5.10.1 Cream.h

```
/* ------------------------------------------------------------------------
| Workfile : Cream.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef CREAM_H
#define CREAM_H

#include <string>

#include "Pricelist.h"
#include "Ingredient.h"

class Cream : public Ingredient {
  std::string GetName() override;
  double GetCost() override;
};

#endif //!CREAM_H
```

5.10.2 Cream.cpp

```
/* ------------------------------------------------------------------------
| Workfile : Cream.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "Cream.h"

using namespace pricelist;

std::string Cream::GetName() {
  return "Cream";
}
double Cream::GetCost() {
  return ingredients::cream;
}
```

5.11 SojaMilk

5.11.1 SojaMilk.h

```
/* ------------------------------------------------------------------------
| Workfile : SojaMilk.h
| Description : [ HEADER ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */


#ifndef SOJAMILK_H
#define SOJAMILK_H

#include <string>

#include "Pricelist.h"
#include "Ingredient.h"

class SojaMilk : public Ingredient {
  std::string GetName() override;
  double GetCost() override;
};

#endif //!SOJAMILK_H
```

5.11.2 SojaMilk.cpp

```
/* ------------------------------------------------------------------------
| Workfile : SojaMilk.cpp
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#include "SojaMilk.h"

using namespace pricelist;

std::string SojaMilk::GetName() {
  return "Soja-Milk";
}
double SojaMilk::GetCost() {
  return ingredients::sojaMilk;
}
```

5.12 PriceList

5.12.1 Pricelist.h

```
/* ------------------------------------------------------------------------
| Workfile : Pricelist.h
| Description : [ SOURCE ]
| Name : Viktoria Streibl     PKZ : S1810306013
| Date : 06.01.20
| Remarks : -
| Revision : 0
| ------------------------------------------------------------------------ */

#ifndef PRICELIST_H
#define PRICELIST_H

namespace pricelist {

  namespace coffee {
    const double blackcoffee = 1;
    const double mocca = 1.20;
    const double espresso = 1;
    const double decaffeinated = 0.8;
  }

  namespace ingredients {
    const double milk = 0.25;
    const double sugar = 0.1;
    const double sojaMilk = 0.3;
    const double cream = 0.5;
  }
}

#endif //!PRICELIST_H
```

5.13 TestDriver

```
1 #include <iostream>
2
3 #include "CoffeePreparation.h"
4
5
6 int main() {
7   std::cout << "Hello World!\n";
8 }
```