НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

## МАГИСТЪРСКИ ФАКУЛТЕТ

# ДЕПАРТАМЕНТ "ИНФОРМАТИКА"

**ПРОГРАМА „Софтуерни технологии в Интернет”**



### Курсов проект

**INFM169 Проект по обектноориентирано програмиране в Интернет**

**ТЕМА: Фризьорски салон**

**НА СТУДЕНТА** : Георги Иванов Георгиев

**Фак. №:** F88683

**Дата:** 06/02/2018 г.

**Преподавател:**

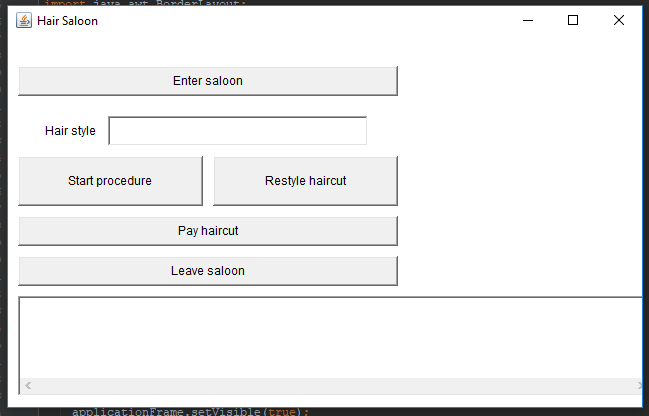
/ проф. д-р. И.Момчев / …………………

София

1. Въведение

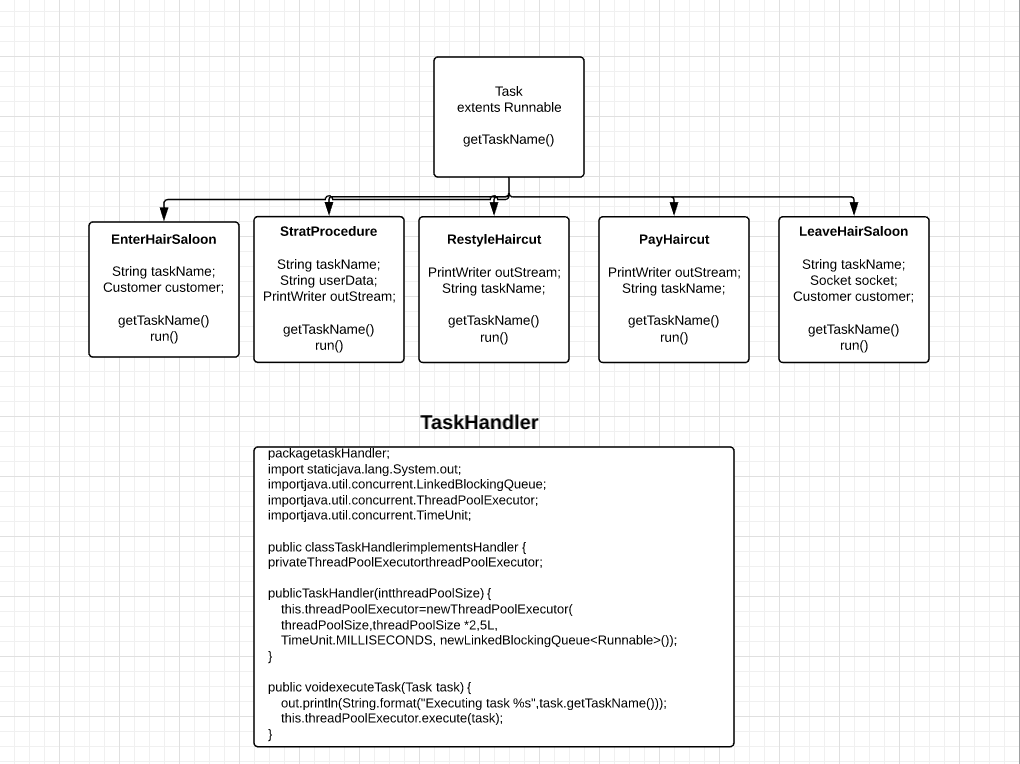
Курсовата работа „Фризьорски салон“ представлява приложение демонстриращо работата на клиент-сървър постановката с реализация на вътрешен оркестратор на задачи и борба за ресурси. Клиентът е поставен в позиция да избира вход в салона и по този начин за него започва работата на приложението. Изборът на бутона вход генерира задача, която се поема от оркестратора и се изпълнява при първа възможност. Веднъж влязъл клиентът въвежда вид фризура в предназначеното за това текстово поле. Това отново генерира задача и тя отново се изпълнява от оркестратора. Информацията за избраната фризура преминава през сокета отворен от сървъра и ако стринга дефиниращ вида фризура не е празен, задачата се изпълнява определено случайно избрано време. След изпълнение на първоначалната фризура, клиентът е поставен през избора да оправи прическата или да я приеме и да плати. Изборът на бутона за повторно стилизиране води до изпълнение на нова задача случайно генерирано време. Изборът на бутона за сметка генерира задача, както и изборът на бутона за изход. Специфичното при изход е освобождаване на мястото заето от клиента.

1. Представяне на потребителския интерфейс



Вече беше споменато, че всеки бутон от потребителския UI генерира задача представена чрез обект имплементиращ интерфейса Task.

Йерархията на класовете имплементирали интерфейса е следната:



TaskHandler e реализация на threadPool, който приема за изпълнение задача от тип Task и я изпълнява, когато има свободна нишка за това.

В текстовото поле се изписват съобщения за потребителя използващ графичния интерфейс. Възможните съобщения са:

* Client is waiting to enter the saloon– клиентът чака да се освободи място в салона. Задачата за вход се изпълнява от оркестратора, но няма свободни места.
* RealCustomer entered the saloon– задачата за вход е приключила изпълнение.
* Please describe your desired haircut– клиентът не е посочил валиден стринг за избрана прическа.
* Working on the haircut – задачата за прическа се изпълнява.
* Client haircut style is: " + userOder + " If you like your hairstyle you can proceed to payment. Else you can restyle your hair.– задачата за изпълнение на прическа е приключила. Потребителят може да рестилизира или да плати.
* Restyling haircut – задачата за рестилизиране се изпълнява.
* Customer haircut has been restyled. Please pay your service. – задачата за рестилизиране е приключила. Клиентът следва да плати сметка.
* Waiting to pay for service. – клиентът изчаква освобождаване на нишка за изпълнение на задача по плащане.
* Customer has paid for the haircut – задачата за плащане е приключила. Клиентът следва да напусне салона.
* Leaving saloon. – клиентът е започнал задача по напускане.
* RealCustomer left the saloon – клиентът е напуснал салона и е освободил място.

1. Описание на класове
   1. Пакет app – съдържа клас HairSaloonApplication, който стартира клиентското приложение.
   2. Пакет common – съдържа класове Customer (обособява наличието на клиент на сървъра) и Entrance (съдържа основни задачи свързани с ресурса място в салона).
   3. Пакет hairSaloon – съдържа клас HairSaloonServer, който стартира сървъра.
   4. Пакет taskHandler – съръжа класове EnterHairSaloonTask, LeaveHairSaloonTask, PayHaircutTask, RestyleHaircutTask, StartProcedureTask, TaskHandler и интерфейсите Handler и Task.
   5. Пакет ui – съдържа класовете EnterL, HairSaloonApplet, LeaveSaloonL, PayHaircutL, RestyleHairCutL и StartProcedureL.
2. Текст на програмата – може да бъде открит на адрес: <https://github.com/Georgegig/nbu-hair-saloon>
3. Инструкция за инсталация

Source кодът се сваля от горепосоченият адрес. При наличието на подходяща среда за разработка (IDE) се стартира файл HairSaloonServer и след това желан на брой пъти се стартира файл HairSaloonApplication.

1. Инструкция за експлоатация

Потребителското поведение трябва да следва следната последователност:

1. Потребителят натиска бутона за влизане в салона и изчаква своя ред;
2. Потребителят въвежда желаната прическа и избира бутона за започване на процедура;
3. Потребителя преценява дали резултата от прическата е задоволителен. Ако е натиска бутона плати сметка. Ако не е натиска бутона рестилизирай;
4. В случай на рестилизиране потребителя изчаква то да приключи, след което заплаща своята сметка;
5. Потребителят натиска бутона напусни салона и напуска салона. Сокет връзката се прекратява и клиентското приложение се затваря.
6. Защо да се използва приложението

Приложението демонстрира базова клиент-сървър постановката. Графичният интерфейс е лесен за реализация. Изпълнението на задачите генерирани от бутоните осъществява идеята на threadPool execution. Условието за борба за място в салона от друга страна илюстрира работата със синхронизиране на нишки. Простотата на приложението би позволила лесното навлизане в основите на многонишковото и клиент-сървърното програмиране с java.