**Финален изпит No. 1**

**Факултен номер на студент:**

**Инструкции:**

1. **Решението на задачите** да се извърши **в този документ**, който **да се предаде** **заедно** с **NetBeans проекта**- решение на задачата за програмиране

2. Решението на **всички задачи** да се **предаде на floppy** и **допълнително** да се качи със **студентския акаунт на Мооdle**.-

3. **Използвайте дадените означения** за *класове*, *променливи* и *методи*.

### *Скала за оценяване:*

### *от 0 до 54 точки*

### *от 55 до 64 точки*

### *от 65 до 74 точки*

### *от 75 до 84 точки*

### *от 85 до 100 точки*

**Забележка**: При установено **преписване** се пише **0 точки за контролното**

**Решете следните задачи** като **спазите изискванията** за **капсулиране**, **скриване на информация** и **многократно използване на код.**

**Задание за програмиране**

Да се **напише приложение на Java** за обработка на фактури като се използва ***Collections*** библиотеката от класове  **в следната последователност** от **задания за програмиране:**

1. **Напишете дефиницията на клас *class Clock* за моделиране на време.** Този клас **има** три данни (*data members*)

* ***Променлива hour – задава час***
* ***Променлива minute –*** ***задава минута***
* ***Променлива second –*** ***задава секунда***

**Напишете:**

* **GET и SET методи за данните на този клас**
* **Конструктор за общо ползване за *class Clock***
* ***toString(),*** който извежда времето във формат **hh:mm:ss AM** или **hh:mm:ss PM** съответно в зависимост от това дали ***hour*** е **по-малък** от **12** или **по- голям** от **12**

**Точки**:20

1. **Напишете дефиницията на *class ClockComparator* ,** който да **е *Comparator* и позволява сравнение на обекти от *class Clock* в низходящ ред на времето по час**, **минути** и **секунди,** съхранявано с ***Clock*** обект**.**

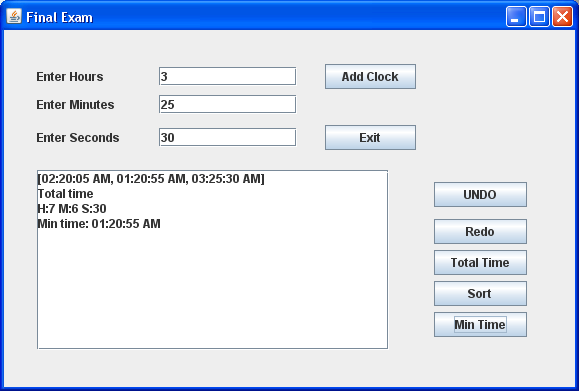
**Точки**:15

1. **Напишете дефиницията на клас *class ClockForm* *,*** който е ***JFrame* приложение** и **има *JPanel* ,** в който е реализиран, **показания по- долу графичен интерфейс**. Този интерфейс съдържа:

* **Етикети** и **текстови** полета (3)
* **Бутони** (7)
* **Текстова** област (1)

Използвайте добре избрани имена за променливите, с които означавате компонентите от графичния интерфейс.

**Точки**:15

******

1. **Нека *class ClockForm* има** инстанции ***baseStack* и *undoStack* на *class Stack*  за работа с *Clock* обекти. Напишете следната обработка на събитие при натискане на бутон**

* **Add Clock *- създава Clock* обект** въз основа на **въведените данни** в текстовите полета на *ClockForm* и *го добавя към baseStack.* Накрая се **извеждат в текстовата област** (*както е показано по- горе*) **списъка с текущите обекти** *Clock* съхраняванив *baseStack*
* **UNDO *- използва undoStack*** за **връщане** на ***baseStack*** в **състоянието му преди последната му промяна** и списъка от елементи на ***baseStack*** в „старото” състояние се **извежда в текстовата област** (*както е показано по- горе*)
* **REDO *- използва undoStack*** за **възстановяване** на ***baseStack*** в **състоянието му преди последното му** UNDOи възстановеният списък от елементи на ***baseStack*** се **извежда в текстовата област** (*както е показано по- горе*)
* **Тоtal Time*- сумира времето,*** показвано **от всеки един** елемент *Clock* на ***baseStack*** и **извежда резултата** по **час**, **минути** и **секунди**(*както е показано по- горе*)
* **Sort*- сортира елементите на* baseStack в низходящ ред на час, минута и секунда като използва *class ClockComparator*** и **извежда елементите на сортирания *baseStack* секунди**(*както е показано по- горе*)
* **Min Time*- намира най- малкият елемент в baseStack*** в съответствие с наредбата дефинирана от ***class ClockComparator*** и **извежда този елемент в текстовата област**(*както е показано по- горе*)
* **Exit*- приключва изпълнението на приложението***

**Точки**:50