

Софийски Университет “Климент Охридски”
Факултет по Математика и Информатика

Контролно No. 1

Курс: Приложно Обектно Ориентирано Програмиране 1

Преподавател: проф. д-р. Е. Кръстев

Студент :

Дата: април 2025

Време за работа: 120 min

Инструкции: Изпълнете следното задание за обектно ориентирано програмиране на Java и предайте изготвения проект на IntelliJ в Мудъл. Пълен набор от точки се присъжда за пълно решение на съответната подзадача. Програмната реализация трябва да спазва концепциите за обектно ориентирано програмиране- encapsulation, information hiding и избягване на дублиране на код.

Оценки:

2	от 0 до 54 точки
3	от 55 до 64 точки
4	от 65 до 74 точки
5	от 75 до 84 точки
6	от 85 до 100 точки

Задача 1 (100 точки)

Задание: Играта [Euchre](#) се играе от двама с тесте от 24 карти. Всяка карта има цвят и сила. Силите на картите са **шест**, подредени в нарастващ ред на стойностите им **1, 2, 3, 4, 5, 6** като **NINE, TEN, JACK, QUEEN, KING, ACE**, а цветовете са четири **HEARTS, SPADES, CLUBS, DIAMONDS**. От всеки цвят има 6 карти, съставени от поредица, където има по една карта от 6-те различни сили. Картите се разбъркват и се теглят последователно извадки от **по 5 карти**. (в последната извадка има по- малко от 5 карти). Играчите натрупват точки в зависимост от поредността на силите на раздадените им извадки от карти. Ако изтеглена карта е коз, нейната сила се умножава по 2 при пресмятане на точките на картите в поредната изтеглена ръка. .

A. Напишете Java приложение на IntelliJ, наречено `CardGameLib`, където създайте `package game` със следното съдържание :

1. Напишете `enum` тип `Faces`, който има следните полета от данни

`NINE, TEN, JACK, QUEEN, .KING, ACE` .

Нека тези полета описват различните сили на тесте от 24 карти за игра

Точки:2

2. Напишете `enum` тип `Suits` който има следните полета от данни

`CLUBS, DIAMONDS, HEARTS , SPADES`

Нека тези полета описват различните цветове на тесте от 24 карти за игра

Точки:2

3. Напишете клас `Card`, който описва една карта от тестето с карти. Добавете в клас `Card` следните данни

`Faces` face; // име на Face (сила на карта)

`Suits` suit; // име на Suit (цвят на карта)

Напишете :

- `get` и `set` методи за данните `face` и `suit`
- Конструктор за общо ползване
- Метод `toString()`, който връща `String` с името на силата и цвета на текущо създаваната карта, например, *KING of HEARTS*

Точки:6

4. Напишете клас *EuchreGame*, който описва тесте от 24 карти и дефинира основни методи за игра с картите. Добавете в клас *EuchreGame* следните данни:

```
Card[] cards; // масив от всички карти в тестето от карти
Card[] hand; // текущо изтеглена ръка от 5 карти
String TRUMP; // константа на всяка отделна игра
Random random; // генератор на случайни числа
int dealCard; // брой карти изтеглени текущо от cards
```

Добавете **статични** едномерни масиви *faces* и *suits*, които съдържат еднократно съответно всички различни сили (6) и цветове(4), описани в Заданието и представени като полета на изброимите типове *Faces* и *Suits*.

Добавете конструктор по подразбиране, където *cards* се инициализира с всички 24 карти (четири цвята, всеки с по 6 различни сили), *TRUMP* (Коз) е `toString()` представяне на произволно избран елемент на *suits*, а останалите с подразбиращи се стойности да са в съответствие с поставените задачи по- долу.

Точки:12

5. В клас *EuchreGame* напишете следните методи

- a) Добавете метод

```
public void shuffleCards(),
```

който **разбърква** елементите на масива *cards* в случаен ред и **инициализира** на нула *dealCard*.

Точки:5

- b) Добавете метод

```
String printCards(),
```

който **връща** текстовото описание на картите в масива *hand*. Картите да са по **три** на ред (последният ред има по- малко) и текстовото им описание да е разделено със запетай, **без запетая в края на реда**. На **отделен ред да се изведе цвета** на текущия Коз (вижте примера в края на текста).

Точки:8

- c) Добавете метод

```
boolean dealHand(),
```

който връща **true** или **false** в зависимост дали са останали или не карти за изтегляне в `cards`. Ако има карти за изтегляне, те се записват последователно в `hand`. В `hand` се записват 5 карти или по-малко, ако толкова са останали в `cards`.

Точки:6

d) Добавете метод

```
int countHandStrength(),
```

връща точките на картите на `hand` (**сумират се силите на картите, където силите на картите КОЗ се умножават по две**).

Точки:8

e) Добавете метод

```
boolean hasKingAndQueenTrump(),
```

който **връща** `true` или `false` в зависимост дали измежду елементите на `hand` има `KING` и `QUEEN` от цвята, избран за **Коз** (Trump).

Точки:10

f) Добавете метод

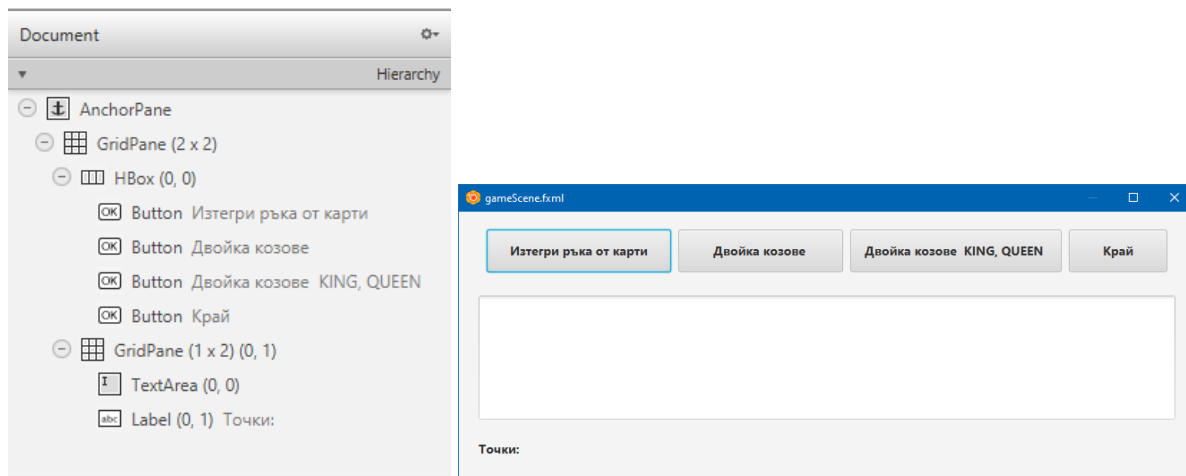
```
boolean hasTwoTrumps(),
```

който **връща** `true` или `false` в зависимост дали измежду елементите на `hand` има двойка карти от избрания цвят за Коз (TRUMP).

Точки:9

B. Създайте потребителски пакет в JAR формат от проекта `CardGameLib` и напишете **нов** IntelliJ проект с **JavaFX приложение** `GameTest`, което **да използва** този потребителски пакет. В така създаденото конзолно приложение създайте `package game` със следното съдържание (Точки 4):

1. Добавете към проекта FXML файл за описание на JavaFX сцена, съответни класове за Контролер и JavaFX приложение (Application). Създайте сцената като използвате следната йерархична структура



Точки:15

2. При инициализацията на Контролера създайте обект от клас *EuchreGame* и изпълнете `shuffleCards()`..

Точки:2

3. Напишете методи за обработка на събитията при натискане на бутоните:

- При натискане на бутона **Изтегли ръка от карти** изпълнете метода `dealHand()` на *EuchreGame* и изведете в горната текстова област резултата от метода `printCards()`. Изведете общия брой точки на текущо показваната ръка в етикета под текстовата област като изпълните метода `countHandStrength()`. При липса на карти за теглене се изписва 0.
- При натискане на бутона **Двойка козове KING, QUEEN** изпълнете метода `hasKingAndQueenTrump()` на *EuchreGame* и изведете текстово описание на резултата в долната текстова област, както е показано на примера .
- При натискане на бутона **Двойка козове** изпълнете `hasTwoTrumps()` на *EuchreGame* и изведете текстово описание на резултата в долната текстова област, както е показано на примера.
- При натискане на бутона **Край** изпълнете команда за прекратяване на JavaFX приложението.

Точки: 12

Примерна реализация

