

**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:** май 2020

**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, inheritance, polymorphism 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 [Bezique](#) се играе от двама с тесте от 64 карти. Всяка карта има цвят и сила. Силите на картите са осем, подредени в намаляващ ред на стойностите им като **ACE, KING, QUEEN, JACK, TEN, NINE, EIGHT, SEVEN**, а цветовете са **CLUBS, DIAMONDS, HEARTS, SPADES**. От всеки цвят има 16 карти, съставени от две поредици, всяка от които има по една карта от 8-те различни сили. Картите се разбъркват, делят се на две тестета и се теглят последователно извадки от по 14 карти като се започва от тестето с по- малко карти. (в последната извадка има по- малко от 14 карти). Когато картите в първото тесте се изчерпят, картите се теглят от второто тесте. Играчите натрупват точки в зависимост от поредността на силите и цветовете на раздадените им извадки от карти .

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

1. Напишете **enum** тип **Face**, който **има** следните константи

**ACE, KING, QUEEN, JACK, TEN, NINE, EIGHT, SEVEN**

Тези константи описват различните сили в тесте от 64 карти за игра.

**Точки:2**

2. Напишете **enum** тип **Suit**, който **има** следните константи

**CLUBS, DIAMONDS, HEARTS, SPADES**

Тези константи описват различните цветове на тесте от 64 карти за игра

**Точки:2**

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

**int** face; // сила на карта

**int** suit; // цвят на карта

Добавете **публични статични** едномерни масиви *faces* и *suits*, които съдържат еднократно съответно **всички различни сили** (8) и **цветове**(4), описани в Заданието и представени като **константи на изброимите типове** *Face* и *Suit*. **Напишете :**

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

Точки:10

4. **Напишете** клас *BeziqueCards*, който **описва** тесте от 64 карти и **дефинира** основни методи за игра с картите. Добавете в клас *BeziqueCards* следните данни:

```
Card[] cards; // масив от всички карти в тестето от карти
Card[] hand;  // текущо изтеглена ръка от най- много 14 карти
Suit trump;   // цвят, избран за Коз на играта
Random random; // генератор на случайни числа
int currentCard; // брой карти изтеглени текущо от cards
```

Добавете конструктор по подразбиране, където *cards* се инициализира с всички 64 карти(четири цвята, всеки с по 8 двойки различни сили), *trump* е произволно избран елемент на *Card.suits*, а останалите с подразбиращи се стойности да са в съответствие с поставените задачи по- долу. **Добавете също** *getTrump()* **метод**

Точки:17

5. В клас *BeziqueCards* **напишете** следните методи

- a) **Добавете** метод

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

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

Точки:5

- b) **Добавете** метод

```
String printCards() ,
```

който връща текстовото описание на картите в масива *hand*. Картите да се описват по **три** на ред (последният ред може да има по- малко) и текстовото им описание да е разделено със запетай.(вижте примера в края на текста) .

Точки:8

с) 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 14 карти или по-малко, ако толкова са останали в cards.

Точки:6

d) Добавете метод

```
boolean hasMarriage( ),
```

връща true или false в зависимост дали измежду елементите на hand има двойка KING и QUEEN от цвета на избрания Коз (trump).

Точки:10

e) Добавете метод

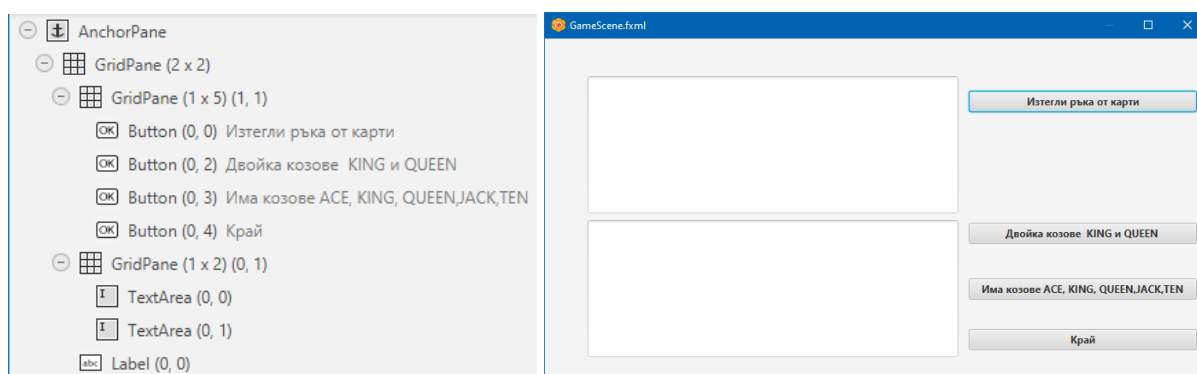
```
boolean hasAKQJ10( ),
```

който връща true или false в зависимост дали измежду елементите на hand има ACE, KING, QUEEN, JACK, TEN от избрания цвят за Коз (trump).

Точки:10

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 view със следното съдържание (Точки 8):

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:2

2. При инициализацията на Контролера създайте обект от клас *BeziqueCards* и изпълнете *shuffleCards()*. В етикета над текстовите области изведете текущо избрания Коз.

Точки:4

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

- При натискане на бутона **Изтегли ръка от карти** изпълнете метода **dealHand()** на **BeziqueCards** и изведете в горната текстова област резултата от метода **printCards()**. При липса на карти, бутонът да се деактивира.
- При натискане на бутона **Двойка Козове KING и QUEEN** изпълнете метода **hasMarriage()** на **BeziqueCards** и изведете текстово описание на резултата в долната текстова област, както е показано на примера ..
- При натискане на бутона **Има козове ACE, KING, QUEEN, JACK, TEN** изпълнете метода **hasAKQJ10()** на **BeziqueCards** 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:16

### Примерна реализация

