

```

public class Card {
    private String rank;
    private String suit;

    public Card(String rank, String suit){
        setRank(rank);
        setSuit(suit);
    }

    public String getRank() {
        return rank;
    }

    public void setRank(String rank) {
        // can only contain legal rank
        if (rank.equals("2") || rank.equals("3") || rank.equals("4")
            || rank.equals("5") || rank.equals("6") || rank.equals("7")
            || rank.equals("8") || rank.equals("9") || rank.equals("10")
            || rank.equals("j") || rank.equals("q") || rank.equals("k")
            || rank.equals("A")) {
            this.rank = rank;
        } else {
            System.out.println("Wrong Rank input!!");
            System.exit(0);
        }
    }

    public String getSuit() {
        return suit;
    }

    public void setSuit(String suit) {
        if (suit.equals("spades") || suit.equals("heart")
            || suit.equals("diamond") || suit.equals("club")) {
            this.suit = suit;
        } else {
            System.out.println("Wrong suit input!!");
            System.exit(0);
        }
    }
}

```

} Rank กับ Suit → ไม่ได้อีกต้องเป็น String (แต่ test ด้วย JUnit ไม่ค่อย)

จะใช้ type อะไรก็ได้ แต่ในเงื่อนไขที่กำหนดมาให้ใช้ไม่ได้

} set ได้เฉพาะ ที่กำหนดมา  
ถ้าใส่ค่าอื่นจะออก error ให้พิมพ์ค่าที่ออก

} set ได้เฉพาะ ที่กำหนดมา  
ถ้าพิมพ์ค่าอื่นจะออก error ให้พิมพ์ค่าที่ออก

```

public int combineRank(Card a) {
    // convert our rank to int
    int first = 0;

    if (rank.equals("j") || rank.equals("q") || rank.equals("k")) {
        first = 10;
    } else if (rank.equals("A")) {
        first = 11;
    } else {
        first = Integer.parseInt(rank);
    }
}

```

→ มีค่าเป็น int  
→ มีทั้งเป็น Card

ให้ this

} J, K, Q ตัดค่าเป็น 10

} 100 A ให้เป็น 11

} แอดให้ค่าการคำนวณแบบธรรมดา

ให้ที่รับเข้ามา

```

// then work out the rank of the second card
int second = 0;
String rankOfA = a.getRank();

```

```

if (rankOfA.equals("j") || rankOfA.equals("q") || rankOfA.equals("k")) {
    second = 10;
} else if (rankOfA.equals("A")) {
    second = 11;
} else {
    second = Integer.parseInt(rankOfA);
}

```

} J, K, Q เป็น 10  
} 100 A ให้เป็น 11  
} แอดให้ค่าการคำนวณแบบธรรมดา

นำผลรวม

```

// combine
int result = first + second;
if (result == 22) {
    result = 12; // if we have 2 "A" we will have to reduce one of them to 1
}
return result;

```

→ คำนวณ

แต่ถ้ามี A สองตัว ให้ตัดค่าเป็น 11 กับ 1 (อย่าเอาผลรวมที่ใน 22 11 กับ 11 ใน แอดให้ค่าการคำนวณด้วยผลรวม 10)

↓  
มีค่า return  
OK

```
Deck.java
public class Deck {
    private Card[] cards;
```

นี่คือวิธีเขียน Deck Card ที่สามารถเล่น 4 type ของไพ่ได้ ✓

```
public Deck() {
    cards = new Card[52];
    String currentRank = "2";
    String currentSuit = "spades";
    int currentIndex = 0;

    for (int i = 1; i <= 13; i++) { // for each rank
        for (int j = 1; j <= 4; j++) { // for each suit
            cards[currentIndex++] = new Card(currentRank,
                currentSuit = nextSuit(currentSuit);
            }
            currentRank = nextRank(currentRank);
        }
    }
}
```

ตอนสร้าง มีอะไรเรียงกัน  
 2 spade, 2 heart,  
 2 diamond, 2 club;  
 3 spade, ---, A club  
 ...  
 2 spade, 3 spade, 4 spade,  
 ---, A spade, 2 heart,  
 3 heart, ---, A club  
 ก็ได้ ✓

```
public Card[] getCards() {
    return cards;
}

public static String nextRank(String s) {
    int temp;
    if (s.equals("10")) {
        return "J";
    } else if (s.equals("J")) {
        return "Q";
    } else if (s.equals("Q")) {
        return "K";
    } else if (s.equals("K")) {
        return "A";
    } else if (s.equals("A")) {
        return "2";
    } else {
        temp = Integer.parseInt(s);
        temp++;
        return String.valueOf(temp);
    }
}
```

หา rank ที่จะเรียงไพ่ต่อไป  
 ได้ทุกค่า

```
public static String nextSuit(String s) {
    if (s.equals("spades")) {
        return "heart";
    } else if (s.equals("heart")) {
        return "diamond";
    } else if (s.equals("diamond")) {
        return "club";
    } else {
        return "spades";
    }
}
```

หา Suit ที่จะเรียงไพ่ต่อไป  
 ได้ทุกค่า

```
public static int random(int first, int last) {
    return first + (int) (Math.random() * (last - first));
}

public static int random(int n) {
    return random(0, n);
}
```

```
public void shuffle() {
    int temp;
    Card tempCard;
    for (int i = 0; i < 52; i++) {
        temp = random(0, 52);
        tempCard = cards[i];
        cards[i] = cards[temp];
        cards[temp] = tempCard;
    }
}
```

สลับไพ่ 9 ครั้ง  
 ทำจนกระทั่งไพ่สลับได้

ถ้าหาไพ่จาก deck ใหม่สำหรับไพ่ที่ต้อง  
ให้คนเล่นด้วย

Deck.java

```

}
}
public card pick() {
    Card result = cards[0];
    if (result == null) {
        System.out.println("No more cards, exit.");
        System.exit(0);
    }

    Card[] newCards = new Card[52];
    int newCardsIndex = 0;

    // copy from the 2nd element onward
    for (int i = 1; i <= 51; i++) {
        newCards[newCardsIndex] = cards[i];
        newCardsIndex++;
    }

    cards = newCards;
    return result;
}

public void cut() { //not used in this lab
    int n_cards = random(0, 51);
    Card[] newCards = new Card[52];
    int newCardsIndex = 0;

    // copy from the nth element onward
    for (int i = n_cards; i <= 51; i++) {
        newCards[newCardsIndex] = cards[i];
        newCardsIndex++;
    }

    // copy the first part to new array
    for (int i = 0; i < n_cards; i++) {
        newCards[newCardsIndex] = cards[i];
        newCardsIndex++;
    }

    cards = newCards;
}
}

```

วิธีนี้คือการ  
ทำในไพ่ใบต่อไป  
จนหมดในมือ  
ให้ Deck  
ที่ต้องในมือ  
ต่อไป  
วิธี

ในกรณี โดนสุ่มจาก mahn → ถ้าเป็นไพ่ของ mahn ก็ให้ไพ่

```

public void playCard(Deck p2) {
    Card p1FirstCard, p2FirstCard, p1SecondCard, p2SecondCard;
    int p1Rank, p2Rank;
    boolean end = false;

    while (!end) {
        p1FirstCard = pick();
        p1SecondCard = pick();
        p2FirstCard = p2.pick();
        p2SecondCard = p2.pick();

        p1Rank = p1FirstCard.combineRank(p1SecondCard);
        p2Rank = p2FirstCard.combineRank(p2SecondCard);

        if (p1Rank > p2Rank) {
            System.out.println("P1 wins.");
            end = true;
        } else if (p2Rank > p1Rank) {
            System.out.println("P2 wins.");
            end = true;
        } else {
            System.out.println("Draw - this round.");
        }
    }
}

```

แต่ละคน หยิบไพ่จาก กองไพ่ 2 ใบ

ถ้าไพ่คนไหนมีค่า  
สูงกว่า

ถ้าไพ่ทั้งสองคนมีค่า  
เท่ากัน

ใน mahn

```

public static void main(String[] args) {
    Deck player1 = new Deck();
    Deck player2 = new Deck();
}

```

```
Deck.java
player1.shuffle();
player2.shuffle(); } สลับทั้ง 2 ครั้ง
//player1.cut();
//player2.cut();

player1.playCard(player2);
```

```
}*
```

```
}
```