# Inheritance Examples Using the PlayingCard Class

### **UML Class Diagram Overview**

#### PlayingCard

value : intsuit : Stringcolor : String

+ PlayingCard(int value, String suit, String color)

+ getValue() : int + getSuit() : String

+ getColor : String

+ setValue(int value) : void

+ setSuit(String suit) : void

+ setColor(String color) : void

+ equals(Object obj) : boolean

+ toString(): String

+ clone(): PlayingCard

+ showCard(): void

#### Legend

# ClassName Data Members format: access name : type access: + = public - = private # - protected / - derived underlined = static

#### **Method Members**

format:

access name(parameters) : return-type

# PlayingCard Class

#### PlayingCard

value : intsuit : Stringcolor : String

- + PlayingCard(int value, String suit, String color)
- + getValue(): int
- + getSuit() : String
- + getColor : String
- + setValue(int value) : void
- + setSuit(String suit) : void
- + setColor(String color): void
- + equals(Object obj) : boolean
- + toString() : String
- + clone(): PlayingCard
- + showCard(): void

#### A PlayingCard has:

- A value
- A suit
- A color
- A 3-arg constructor that takes a value, suit and color
- The standard getters and setters
- A showCard() method to display the attributes
- A toString() method to convert the values to a String
- An equals() method to determine if two PlayingCard objects have the same value, suit and color
- A clone() method to make a copy of a PlayingCard

## American Playing Card Class

#### **PlayingCard**

value : intsuit : Stringcolor : String

- + PlayingCard(int value, String suit, String color)
- + getValue(): int
- + getSuit() : String
- + getColor : String
- + setValue(int value) : void + setSuit(String suit) : void
- + setSuit(String suit) : void + setColor(String color) : void
- + equals(Object obj) : boolean
- + toString(): String
- + clone() : PlayingCard
- + showCard(): void



#### **AmericanPlayingCard**

- DEFAULTCARDVALUE: int=0
- DEFAULTCOLOR: String="BLACK"
- DEFAULTSUIT : String="Joker"
- MAXVALUE int=13
- MINVALUE : int=0
- suitMap : Map<String, String>
- valueMap : Map(Integer, String>
- + AmericanPlayingCard()
- + AmericanPlayingCard(int value, String suit) : void
- + toString() : String
- + clone(): PlayingCard
- initializeMaps(): void

#### An AmericanPlayingCard is a PlayingCard with:

- Default value of 0
- Default suit of "Joker"
- Default color of "Black"
- A maximum value of 13
- A minimum value of 0
- A suitMap() used to validate suits and colors
  - Suits/Colors are: SPADES/BLACK, CLUBS/BLACK, HEARTS/RED, DIAMONDS/RED
- A valueMap to convert values to corresponding String
- A 2-arg constructor that takes a value, suit
- The standard getters and setters
- A clone() method to make a copy of an AmericanPlayingCard
- An initializeMaps() method to initialize the suitMap and valueMap with the valid values

# **Italian** (Scopa) **Playing Cards**



- Values:
  - 1 to 7, 10, "Fante", "Cavallo", "Re"
- Suits/Colors are:
  - COINS Yellow
  - CUPS Blue
  - SWORDS Red
  - BATONS Black

# ItalianPlayingCard Class

#### **PlayingCard**

- value : int
- suit : String - color : String
- + PlayingCard(int value, String suit, String color)
- + getValue(): int
- + getSuit(): String
- + getColor : String
- + setValue(int value) : void
- + setSuit(String suit) : void
- + setColor(String color): void + equals(Object obj): boolean
- + toString(): String
- + clone(): PlayingCard
- + showCard(): void



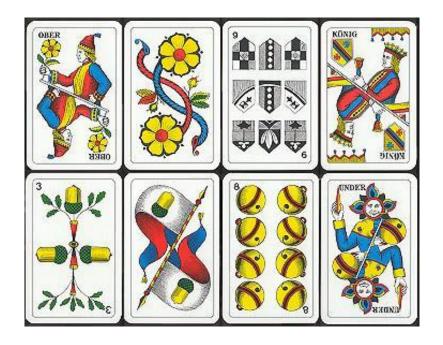
#### **ItalianPlayingCard**

- DEFAULTCARDVALUE : int=0
- DEFAULTCOLOR: String="BLACK"
- DEFAULTSUIT : String="Joker"
- MAXVALUE int=13
- MINVALUE: int=0
- suitMap : Map<String, String>
- valueMap : Map(Integer, String>
- + ItalianPlayingCard()
- + ItalianPlayingCard(int value, String suit)
- + toString(): String
- + clone(): PlayingCard
- initializeMaps(): void

#### An ItalianPlayingCard is a PlayingCard with:

- Default value of 0
- Default suit of "Joker"
- Default color of "Black"
- A maximum value of 13 Note: There are no 8's or 9's
- A minimum value of 0
- A suitMap() used to validate suits and colors
  - Suits/Colors are: COINS/YELLOW, CUPS/BLUE, SWORDS/RED. BATONS/BLACK
- A valueMap to convert values to corresponding String
  - 11="Fante", 12="Cavallo", 13="Re"
- A 2-arg constructor that takes a value, suit
- The standard getters and setters
- A clone() method to make a copy of an ItalianPlayingCard
- An initializeMaps() method to initialize the suitMap and valueMap with the valid values

# **Swiss Playing Cards**



#### Values:

- 6 to 9, "Banner", "Under", "Ober", "König"; 1 "As"
- Suits/Colors are:
  - BALLS Yellow
  - o ACORNS Green
  - o ROSES Red
  - SHIELDS Black

# SwissPlayingCard Class

#### **PlayingCard**

- value : int - suit : String
- color : String
- + PlayingCard(int value, String suit, String color)
- + getValue(): int
- + getSuit(): String
- + getColor : String
- + setValue(int value) : void
- + setSuit(String suit) : void + setColor(String color): void
- + equals(Object obj): boolean
- + toString(): String
- + clone(): PlayingCard
- + showCard(): void

#### SwissPlayingCard

- DEFAULTCARDVALUE: int=0
- DEFAULTCOLOR: String="BLACK"
- DEFAULTSUIT : String="Joker"
- MAXVALUE int=13
- MINVALUE: int=6
- suitMap : Map<String, String>
- valueMap : Map(Integer, String>
- + SwissPlavingCard()
- + SwissPlayingCard(int value, String suit)
- + toString(): String
- + clone(): PlayingCard
- initializeMaps(): void

An SwissPlayingCard is a PlayingCard with:

- Default value of 0
- Default suit of "Joker"
- Default color of "Black"
- A maximum value of 13
- A minimum value of 6 (except 1 is allowed))
- A suitMap() used to validate suits and colors
  - Suits/Colors are: BALLS/YELLOW, SHIELDS/BLACK. ROSES/RED. ACORNS/GREEN
- A valueMap to convert values to corresponding String
  - 1="As, 10="Banner", 11="Under", 12="Ober", 13="König"
- A 2-arg constructor that takes a value, suit
- The standard getters and setters
- A clone() method to make a copy of an SwissPlayingCard
- An initializeMaps() method to initialize the suitMap and valueMap with the valid values