

Chair of Software Engineering



# Einführung in die Programmierung Introduction to Programming

Prof. Dr. Bertrand Meyer

Exercise Session 9

## **Today**



> Feedback on the mock exam

> Exercise: practicing contracts

## Specification of a card game



A deck is initially made of 36 cards

Every card in the deck represents a value in the range 2..10

Every card also represents 1 out of 4 possible colors

The colors represented in the game cards are: red ('R'), white ('W'), green ('G') and blue ('B')

As long as there are cards in the deck, the players can look at the top card and remove it from the deck

#### **Class CARD create make**

make (a\_color: CHARACTER, a\_value: INTEGER)

-- Create a card given a color and a value.

require

• • •

ensure

• • •

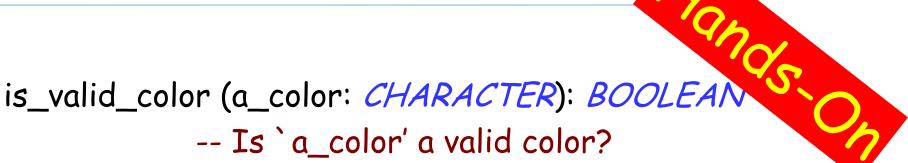
color: CHARACTER

-- The card color.

value: INTEGER

-- The card value.

## Class CARD: which colors are valid?



require

• • •

ensure

• •

## Class CARD: which ranges are valid?

```
is_valid_range (n: INTEGER): BOOLEAN
-- Is `n' in the acceptable range?
require
```

• • •

ensure

• •

invariant

• • •

#### Class CARD create make: reloaded

make (a\_color: CHARACTER, a\_value: INTEGER)

-- Create a card given a color and a value.

require

• • •

ensure

• • •

color: CHARACTER

-- The card color.

value: INTEGER

-- The card value.

#### **Class DECK create make**

# m cards.

#### make

-- Create a deck with random cards.

require

• • •

ensure

• •

feature {NONE} - Implementation

card\_list: LINKED\_LIST

-- Deck as a linked list of cards.

### **Class DECK queries**

```
top_card: CARD
            -- The deck's top card.
is_empty: BOOLEAN
            -- Is Current deck empty?
      do
      end
count: INTEGER
            -- Number of remaining cards in the deck.
      do
      end
```

## Removing the top card from DECK



remove\_top\_card

-- Remove the top card from the deck.

require

• • •

ensure

• •

#### The class invariant



#### invariant

• • •