Bluegrass Community and Technical College

Programming Requirements Document

Dalmuti Deal - Revisited

Narrative Description

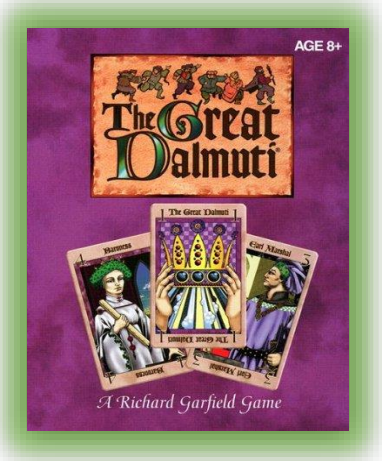
We are taking a break from classes this week to learn about simple arrays. We will also practice using static methods in our programs.

In a previous week using a non-standard 80-card deck, you worked with selection statements and repetition structures to create a card deck for a game called ***The Great Dalmuti***. You displayed the cards on the screen, dealt two random cards, and determined which card was higher (or possibly a tie).

This week you will create an 80-element array to hold the cards from ***The Great Dalmuti*** deck. Afterwards, you will ask the user how many players are in the game, and you will deal all the cards in the deck to the appropriate players.

As a reminder, here are the cards in the deck:

The Great Dalmuti – Card Layout		
Rank of Card	Name of Card	Count of this type of card in a deck
1	Dalmuti	1
2	Archbishop	2
3	Earl Marshal	3
4	Baroness	4
5	Abbess	5
6	Knight	6
7	Seamstress	7
8	Mason	8
9	Cook	9
10	Shepherdess	10
11	Stonecutter	11
12	Peasant	12
13	Jester	2



For this assignment, you are not programming the entire game but rather you creating and dealing the cards using a single-dimension array.

Step 1: Create and Populate an Array to Hold the Dalmuti Cards

Generate strings for each of the 80 cards and load the strings into the array. The string will be constructed by concatenating

(1) The rank of the card,
(2) a colon and space, and
(3) the name of the card

Examples of properly formatted strings are:

1: Dalmuti
2: Archbishop
3: Earl Marshal
4 Baroness
…
12: Peasant
13: Jester

Remember there are multiple copies of cards to form the 80-card deck.

Step 2: Shuffle the Cards using a Static Method

You read about static methods in Chapter 6 of zyBooks this week. You read about common array algorithms, such as shuffling, this week in the handout entitled, *Array Summary*.

Using a static method named **shuffleCards(),** shuffle the cards in the array by randomly swapping elements in the array. HINT: Code for shuffling a deck of card was given in the *Array Summary* handout.

Step 3: Deal the Cards using a Static Method

The game can be played with 4-8 players. However, we will assume the game is being played with 4 players. After the cards have been shuffled in the deck (moved around in the array), deal all the cards to the players. Cards are dealt one at a time from the top of the deck until all cards are given to the players.
Display the card in the following format:

Player1	Player2	Player 3	Player4
4: Baroness	2: Archbishop	12: Peasant	6: Knight
10: Shepherdess	11: Stonecutter	1: Dalmuti	3: Earl Marshal
…			

Use a static method to deal the cards.

Restrictions:

This week we are temporarily switching our focus to creating and using arrays apart from user-defined classes. Do not use any user-defined classes for this assignment. We will include used-defined classes with arrays in Module 12.

Software Requirements

R1: The user interface (what the use sees and how they interact with the program) is intuitive, clear, and easy to use.
R2: An 80-element single-dimension array is used to hold strings which represent cards from a Dalmuti deck.
R3: The array is properly created as defined and with the 80 cards it should hold.
R4: The cards in the array are "shuffled" by repetitively and randomly swapping cards in the array.
R5: A static method is used to shuffle the cards in the array.
R6: Cards are properly dealt from the top of the deck (beginning of the array) one at a time alternately between players. In other words, player 1 get the first card, player 2 gets the second card, player 3 gets the third card, etc.
R7: The program properly displays each player's cards.
R8: A static method is used to create the display of players' cards.
R9: Previous standard requirements are included (documentation, selection of proper data types, constants, etc.).

Security Considerations

Array out-of-bound errors can cause security vulnerabilities.

Special Notes

None provided for this assignment.

Change Request Form

Students who wish to obtain written permission to alter the assignment or to use features/statements/structures before they are introduced in class, must complete a *Change Request Form* (link in Blackboard in left-hand navigation bar) and follow all guidelines provided there.