# 实验4

## 实验目的：

* 熟悉数组的使用
* 熟悉类的使用

## 实验要求：

* 按照题目要求写代码，并上传到COURSE.XMU.EDU.CN

## 实验题目：

1． *(Game of Craps)* Write an application that runs 1,000,000 games of craps (Fig. 6.8) and answers the following questions:

* a)  How many games are won on the first roll, second roll, ..., twentieth roll and after the twentieth roll?
* b)  How many games are lost on the first roll, second roll, ..., twentieth roll and after the twentieth roll?
* c)  What are the chances of winning at craps? [*Note:* You should discover that craps is one of the fairest casino games. What do you suppose this means?]
* d)  What is the average length of a game of craps?
* e)  Do the chances of winning improve with the length of the game?

2. *(Airline Reservations System)* A small airline has just purchased a computer for its new au- tomated reservations system. You’ve been asked to develop the new system. You’re to write an ap- plication to assign seats on each flight of the airline’s only plane (capacity: 10 seats).

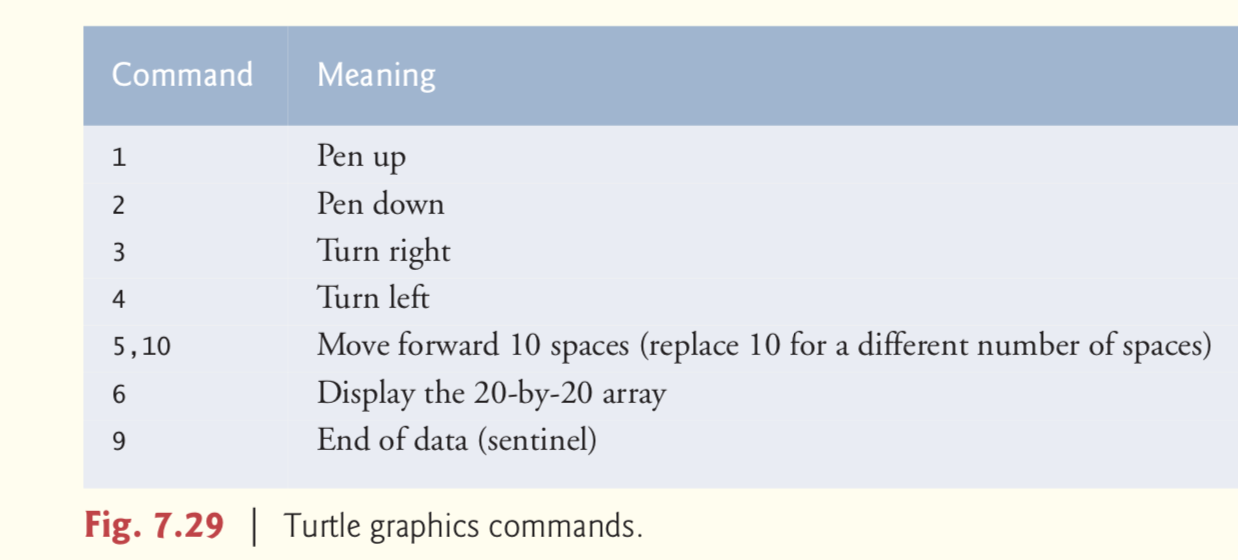
Your application should display the following alternatives: Please type 1 for First Class and Please type 2 for Economy. If the user types 1, your application should assign a seat in the first- class section (seats 1–5). If the user types 2, your application should assign a seat in the economy section (seats 6–10). Your application should then display a boarding pass indicating the person’s seat number and whether it’s in the first-class or economy section of the plane.

Use a one-dimensional array of primitive type boolean to represent the seating chart of the plane. Initialize all the elements of the array to false to indicate that all the seats are empty. As each seat is assigned, set the corresponding element of the array to true to indicate that the seat is no longer available.

Your application should never assign a seat that has already been assigned. When the economy section is full, your application should ask the person if it’s acceptable to be placed in the first-class section (and vice versa). If yes, make the appropriate seat assignment. If no, display the message "Next flight leaves in 3 hours."

3. *(Turtle Graphics)* The Logo language made the concept of *turtle graphics* famous. Imagine a mechanical turtle that walks around the room under the control of a Java application. The turtle holds a pen in one of two positions, up or down. While the pen is down, the turtle traces out shapes as it moves, and while the pen is up, the turtle moves about freely without writing anything. In this problem, you’ll simulate the operation of the turtle and create a computerized sketchpad.

Use a 20-by-20 array floor that’s initialized to zeros. Read commands from an array that con- tains them. Keep track of the current position of the turtle at all times and whether the pen is cur- rently up or down. Assume that the turtle always starts at position (0, 0) of the floor with its pen up. The set of turtle commands your application must process are shown in Fig. 7.29.



Suppose that the turtle is somewhere near the center of the floor. The following “program”

would draw and display a 12-by-12 square, leaving the pen in the up position:

2

5,12

3

5,12

3

5,12

3

5,12

1

6

9

As the turtle moves with the pen down, set the appropriate elements of array floor to 1s. When the 6 command (display the array) is given, wherever there’s a 1 in the array, display an asterisk or any character you choose. Wherever there’s a 0, display a blank.

Write an application to implement the turtle graphics capabilities discussed here. Write several turtle graphics programs to draw interesting shapes. Add other commands to increase the power of your turtle graphics language.

4. *(Enhancing Class Time2)* Modify class Time2 of Fig. 8.5 to include a tick method that in- crements the time stored in a Time2 object by one second. Provide method incrementMinute to in- crement the minute by one and method incrementHour to increment the hour by one. Write a program that tests the tick method, the incrementMinute method and the incrementHour method to ensure that they work correctly. Be sure to test the following cases:

* a)  incrementing into the next minute,
* b)  incrementing into the next hour and
* c)  incrementing into the next day (i.e., 11:59:59 PM to 12:00:00 AM).