DUE DATE  
2021.12.10 금요일 오후 11시 50분

제출일

2021.12.07 화요일

제출자 이름

: 김현욱

제출자 학번

: 21800201

숙제 번호 기록

: REP-Chap#13

문제[Question]

영문버전

Write an application with a GUI that will convert numbers from binary to octal. Binary numbers are composed of just the digits 0 and 1. Octal numbers use the digits 0, 1, 2, 3, 4, 5, 6, and 7. Note that each octal digit corresponds to a three-bit binary number, as follows:

Binary Octal 000 0 / 001 1 / 010 2 / 011 3 / 100 4 / 101 5 / 110 6 / 111 7

To convert a number from binary to octal, first group the bits in the binary number into sets of three and then apply the equivalent octal numbers. For example, the binary number 001000101110 would be grouped as 001 000 101 110, which corresponds to the octal digits 1, 0, 5, and 6, respectively, Thus, the octal equivalent of the binary number is 1056. If the number of bits in the binary number is not divisible by 3, add zeros at its beginning until it is. For example, since the binary number 1011011100100 has 13 bits, we would add two zeros before it to get 001011011100100. We would then group its bits as 001 011 011 100 100 and get 13344 as its octal equivalent

To convert an octal number to binary, we use our correspondence table in the reverse direction. For example, the octal number 716 is 111 001 110, or 111 001 110 in binary. Your application can omit the spaces we use to show the grouping of the bits. You will need a text field for the user to enter a number and a label for the results. Provide three buttons: To Octal, To Binary, and Clear. If the user clicks the Clear button, clear any text in the text field and label. If the user clicks one of the two conversion buttons, check whether the number in the text field is in the correct format. If it is not, display the message Bad Format in the results label. If the format is ok, compute the converted value and display that in the results label

한글 해석

이진수를 십진수로 변경해주는 어플리케이션을 GUI와 같이 작성해라. 이진수 숫자는 0과 1로 이루어져 있다. 십진수 숫자는 0에서 7까지의 숫자를 사용한다. 다음과 같이 십진수 숫자는 3비트의 이진수의 숫자와 상응함을 주목해라.

이진수 십진수 : 000 0 / 001 1 / 010 2 / 011 3 / 100 4 / 101 5 / 110 6 / 111 7

이진수를 십진수로 변환하기 위해서 3개의 bit로 이루어져 있는 것이 하나의 그룹이 되어 십진수에 상응하는 것으로 적용이 된다. 예를 들어 이진수 001000101110는 001 000 101 110 로 그룹이 묶여지는데 각각 1,0,5,6으로 상응한다. 그러므로 이에 해당하는 십진수는 1056이다. 만약 이진수 숫자의 bit 개수가 3으로 나누어 떨어지지 않으면 처음에 0들을 추가한다. 예를 들어 1011011100100은 13개의 비트를 가지므로 001011011100100 처럼 앞에 2개의 0을 추가한다. 이렇게 되면 001 011 011 100 100 으로 그룹화 되고 13344 라는 십진수 숫자를 얻을 수 있다.

십진수를 이진수로 바꾸기 위해서 반대 방향의 상응하는 표를 사용해야 한다. 예를 들어 십진수 716은 111 001 110 또는 111 001 110 이진수 숫자이다. 너의 어플리케이션은 비트의 그룹을 보여주기 위한 공백을 제거할 수 있다. 너는 사용자가 숫자를 입력하고 결과를 보여주기 위한 곳으로써 text field가 필요할 것이다. 3개의 To Octal, To Binary, Clear 버튼을 제공해라. 만약 사용자가 Clear 버튼을 누르면 text field와 label에 있는 것을 삭제해라. 만약 사용자가 2개의 변환 버튼을 누르면 text field에 있는 숫자가 올바른 형식인지 확인하고 아닌 경우 Bad Format 메시지를 출력해라. 만약 올바른 형식이면 변형된 결과를 계산해서 출력해라.

기본 문제 해결 개념

기본적인 윈도우 틀을 만들기 위해서 JPanel과 JLabel, JButton 그리고 GridLayout과 BorderLayout을 적절히 조합하여 원하는 결과값과 최대한 비슷하게 구현할 수 있다. 그리고 십진수를 이진수로 바꾸기 위해서 버튼이 클릭되는 Event가 들어오면 JText에 있는 문자열을 chatAt 메소드를 이용하여 이진수 문자열 변환을 한다. 반대로 이진수를 십진수로 바꾸기 위해서 입력받은 문자열이 3개로 나누어지는 구조인지 파악하는 조건문을 만들어 다룰 수 있는 형태로 만들어준다. 이후 3개의 문자씩 끊어 equals 와 substring 메소드를 이욯해서 원하는 십진수로 출력을 한다. Clear의 경우 셋팅해 놓은 TextField의 setText 메소드를 이용해서 “” 빈 문자열을 할당해 작성된 문자열을 삭제한다.

설계도

이진수와 십진수의 변환을 진행하기 전, 해당하는 윈도우 창을 만들기 위해서 JPanel과 JButton, JLabel, JTextField를 이용해서 비슷한 윈도우 창을 만든다. 좌우로 TextField와 Button이 위치하기 때문에 GridLayout을 큰 틀로 각각의 Button과 Label이 들어가는 곳은 BorderLayout을 사용하는 Panel 안에 담아서 적절한 배치를 한다. 이후 환경 설정이 완료되었으면 변환을 시작한다. 이진수로 변환하기 위해서 문자열의 하나하나 문자에 해당하는 이진수 문자열을 출력해야 한다. 이는 charAt 문자 비교와 setText를 통해 수행할 수 있다. 그리고 십진수로 변환하기 위해서 문자열을 가공하는 과정이 필요하다. 문자열이 0과 1로 이루어진 숫자이며 총 문자열이 3개씩 그룹으로 묶일 수 있어야 하는데 그렇지 못한 경우 앞에 “0” 또는 “00” 을 붙인 새로운 문자열을 만들어 조건에 맞도록 문자열을 가공한다. 이후 substring으로 문자 3개와 equals를 통해 각각 어떤 십진수 숫자에 대응하는지 비교하고 setText를 통해 차례차례 출력한다. 마지막으로 Clear는 TextField 영역과 Label 영역의 글을 없애주어야 하므로 setText에 “” 아무것도 없는 문자열을 할당해 작성된 문자열을 삭제하는 방향으로 전체 코드를 설계하면 될 것이다.

코드 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Converter\_main 테스트 클래스를 실습에서 진행했던 방식 그대로 선언하여 클래스를 만든다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Converter 클래스. WIDTH와 HEIGHT 상수선언을 해준다. Buttonpanel 변수는 오른쪽에 위치할 버튼이 담길 큰 틀에 해당하는 변수로 GridLayout(3,1)을 해주어서 행이 3개 열이 1개에 해당하는 틀을 만들어준다. 그리고 buttonpanel1,2,3는 각각의 버튼에 해당하는 JPanel로 BorderLayout을 해주어 문제에서 주어진 그림과 비슷하게 나오도록 만들어준다. Panel2는 왼쪽에 위치하는 텍스트가 담길 JPanel에 해당하며 이도 마찬가지로 GridLayout(3,1)을 해주어서 행이 3 열이 1에 해당하는 틀을 만들어주고, areapanel1은 결과값이 나오는 JLabel이 들어가며 areapanel2는 해당 변환이 유효한지 무효한지 출력해주는 JLabel이 속하게 된다. 마지막으로 JButton은 각각 이진수 변환, 십진수 변환, Clear에 해당하는 버튼을 만들게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

윈도우 창의 사이즈와 타이틀 이름을 다음과 같이 설정을 한다. 다음으로 TextField를 추가하게 되는데 theText는 JTextField에 해당하는 것으로 BorderLayout.WEST를 하면 문제에서 주어진 상황과 동일하게 출력이 되었음을 확인할 수 있다. 나머지 JLanel은 각각 areapanel1,2 선언한 곳에 SOUTH 에 위치하도록 먼저 add를 해주고 전제척인 큰 틀인 panel2에 add 함으로써 Text 영역은 셋팅을 완료한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Buttonpanel1,2,3에 해당하는 Panel에 각각의 버튼을 BorderLayout.NORTH로 넣어주고 다시 각각의 buttonpanel1,2,3 들을 buttonpanel이라는 큰 틀에 해당하는 JPanel에 넣어준다. 각각의 셋팅이 완료되고 마지막으로 contentPane에 add를 해서 왼쪽의 Text 부분과 오른쪽의 Button 부분의 설정을 완료한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

actionPerformed메소드에서 e.getActionCommand() 를 통해서 어떤 것이 사용되었는지 actionCommand 문자에 남고 각각의 변수들을 선언해준다.

텍스트, 모니터, 실내, 화면이(가) 표시된 사진

자동 생성된 설명

Convert to octal이라는 버튼이 눌려졌다면 다음과 같은 문장을 실행하게 된다. 문자열 arr을 선언함과 동시에 theText에 쓰여진 문자를 받는 getText과 공백을 제거해 주는 replaceAll 메소드를 통해서 초기화를 시켜준다. 우선 해당 문자열이 0과 1이 아닌지 파악하여 아닌 숫자가 있다면 error를 카운트 해주고 해당 for문이 끝난 동시에 error가 0보다 크다면 area2 Label에 Sorry, can not convert that 이라는 문장을 작성하고 그렇지 않은 경우에는 Value converted to octal이라는 문장을 작성하게 해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다음은 3개로 나누어지는 그룹핑이 되는지를 확인해야한다. 만약 문자열의 길이가 3으로 나누어 떨어지지 않으면 해당 조건문을 실행한다. 길이를 3으로 나눈 나머지가 1이라면 문자열 앞에 “00”을 붙인 새로운 문자열을 생성하고 나머지가 2라면 문자열 앞에 “0”을 붙인 새로운 문자열을 생성한다. 만약 3으로 그룹핑이 잘 되었다면 원래의 문자열을 new\_arr에 할당해 해당 조건문을 완성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

십진수 변환은 가공된 문자열 new\_arr를 이용해서 진행하게 된다. New\_arr의 substring 메소드를 이용해서 3개 단위로 문자열을 끊고 그것이 어떤 이진수와 동일한지 비교하는 if 문을 통해서 각각에 해당하는 십진수 숫자를 area1에 출력해주게 된다.

텍스트, 모니터, 화면, 은색이(가) 표시된 사진

자동 생성된 설명

Convert to binary 버튼이 눌린 경우 마찬가지고 theText에 작성된 문자열을 가져다 주는 getText 메소드를 이용하고 replaceAll을 통해 공백을 제거하게 된다. 이진수 변환의 경우 각각의 문자들이 0과 9 사이에 위치하지 않으면 error가 발생했으므로 0보다 작거나 9보다 큰 경우에 error 변수를 증가시켜준다. for문이 끝남과 동시에 error가 0보다 크다면 Sorry, can not convert that을 area2에 출력하고 그렇지 않다면 Value converted to binary를 area2에 출력해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

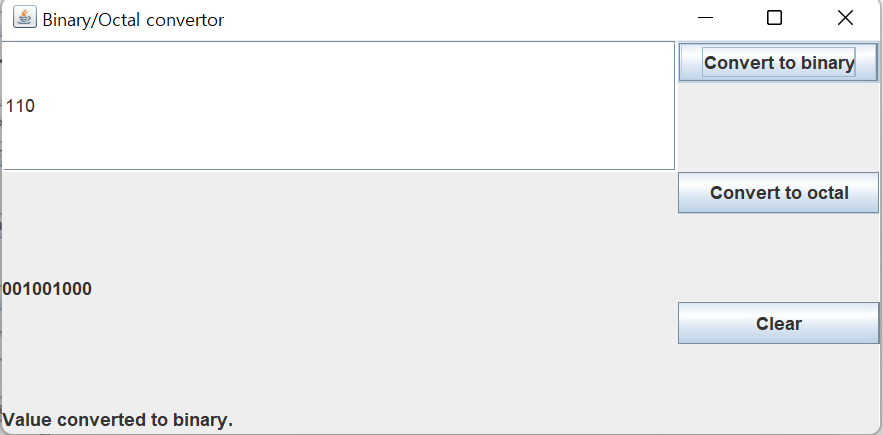
가공된 문자열 arr의 charAt 메소드를 이용하여 각각의 문자가 해당하는 이진수 숫자를 찾고 그것을 area1에 출력해주게 된다.

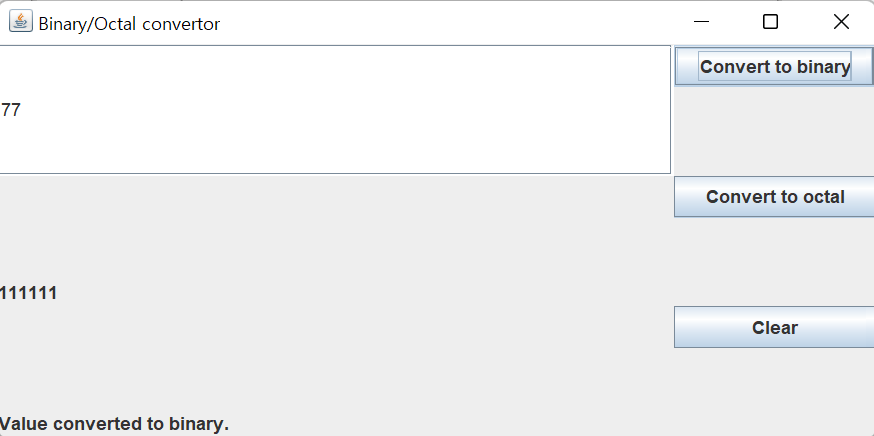
텍스트이(가) 표시된 사진

자동 생성된 설명

Clear 버튼이 눌린 경우 area1, area2, theText에 해당하는 문자열에 setText로 빈 문자열을 할당해 기존의 문자열을 없애주는 효과를 준다.

결과화면





텍스트이(가) 표시된 사진

자동 생성된 설명

