**자바실무 기말고사(실기)**

학번 : 이름 :

201720970

권대한

문제1)

**숫자 야구 게임 작성하기(10점)**

TextField와 입력 버튼**을 생성**한다.

아래쪽에 **스트라이크, 볼, 아웃**을 그래픽 메소드로 처리한다.(**paintComponent 메소드 사용**)

스트라이크는 RED, 볼은 BLUE, 아웃은 GREEN으로 하시면 됩니다.

**컴퓨터 내부에서 중복되지 않는 난수값 3개를 생성하여 사용자가 입력한 값과 비교한다. 난수(1~9까지의 수)**

① 타이틀에 자신의 학번과 이름을 추가하시오. - 타이틀 출력( 야구 GUI 게임(학번 이름) )

② 클래스명 : Gimal9122126Exam – 9122126은 자신의 학번을 의미한다.

③ 화면 디자인

④ 스윙 프레임 작성, 크기 - 400×400 ⑤ 배치 관리자 사용

⑥ JPanel, JTextField, JButton 모두 사용

⑦ Random 메소드 사용(중복되지 않는 난수)

⑧ ActionListener 사용

⑨ 입력 다이얼로그 사용 - JTextField에 출력

⑩ TextField는 문자열 편집을 불가능하게 만든다.

- 난수로 발생시키는 수는 출력하는 부분을 코딩하고 주석처리해주시기 바랍니다.

(채점을 빠르게 하기 위하여)

-. 텍스트와 사각형에 대한 색상 설정은 스트라이크 - RED, 볼 - BLUE, 아웃 - GREEN으로 하시면

됩니다.

-. 실행화면1은 초기화면입니다. JTextFiled, JButton을 추가하시고 그래픽 메소드를 이용하여

텍스트와 사각형을 출력하시면 됩니다.

-. 실행화면2는 스트라이크가 3개인 경우입니다. 폰트를 설정하시고 그래픽 메소드를 이용하여

텍스트를 출력하시면 됩니다.

-. 실행화면3는 스트라이크가 2개인 경우입니다. 그래픽 메소드를 이용하여 텍스트,

사각형을 출력하시면 됩니다.

스트라이크의 개수가 채워진 사각형의 개수와 같습니다.

-. 실행화면4는 볼이 3개인 경우입니다. 그래픽 메소드를 이용하여 텍스트, 사각형을 출력하시면

됩니다.

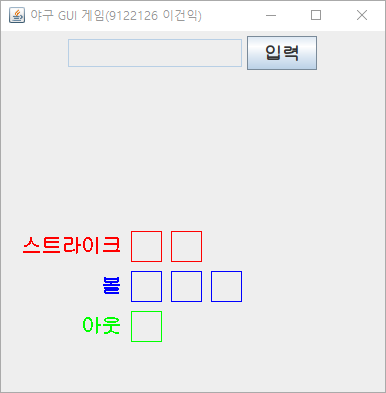
볼의 개수가 채워진 사각형의 개수와 같습니다.

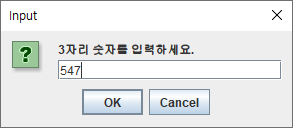
-. 실행화면5는 아웃입니다. 숫자를 하나도 못 맞추었을 경우에는 아웃으로 판단합니다.

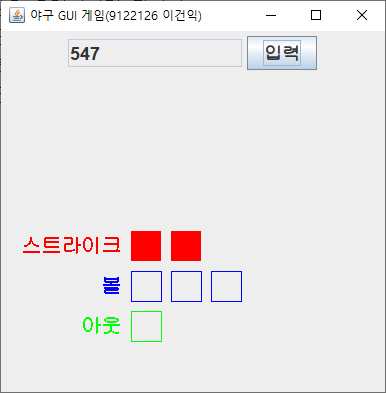
**-. 설명이 없는 부분은 임의로 설정하거나 작성하시면 됩니다.**

[실행화면1] - 초기화면입니다. 입력을 누르면 입력 다이얼로그가 나타나고 3자리를 숫자를 입력하면 텍스트필드에 입력된 3자리 숫자가 출력되면서 스트라이크와 볼, 아웃을 판단합니다.

- 폰트설정은 임의로 하시기 바랍니다.

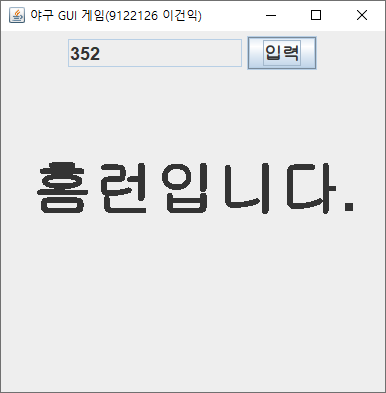






[실행화면2] - 스트라이크가 3개인 경우(“홈런입니다.” 텍스트를 출력)

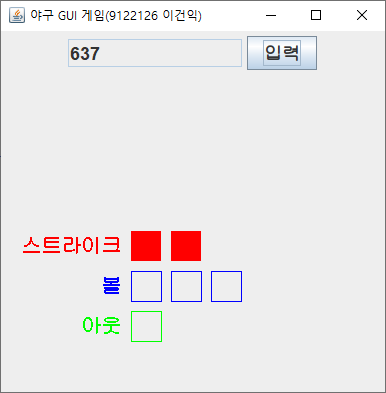
- 폰트설정은 임의로 하시기 바랍니다.



[실행화면3] - 스트라이크가 2개인 경우

- 스트라이크를 도형 칠하기 메소드로 채우시면 됩니다.

**(스트라이크 2개, 볼 0개)**

****

**[실행화면4] - 볼이 3개인 경우**

- 볼을 도형 칠하기 메소드로 채우시면 됩니다.

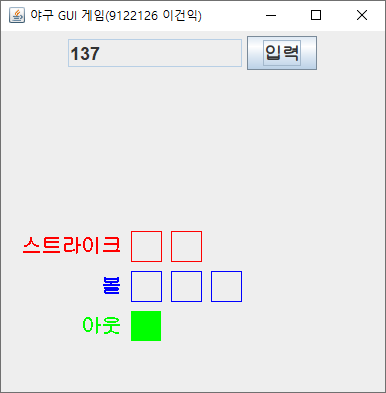
**(스트라이크 0개, 볼 3개)**

****

**[실행화면5] - 아웃입니다. 숫자를 하나도 못 맞추면 아웃입니다.**

- 아웃을 도형 칠하기 메소드로 채우시면 됩니다.

**(아웃)**

****

※ 코딩과 결과화면은 아래쪽에 자유롭게 삽입하시오.

코딩은 반드시 텍스트로 추가하시기 바랍니다.

코딩을 텍스트로 추가하지 않으면 채점이 불가능합니다.

[코딩 추가]

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**public** **class** Gimal201720970Exam **extends** JFrame {

**public** **int**[] GenRandom() { // 요구점인 Random 메소드를 사용하여, 각 메소드에 랜덤 생성 값을 전달하기 위해, 배열 리턴

**int**[] random = **new** **int**[3];

random[0] = (**int**) (Math.*random*() \* 9 + 1);

random[1] = (**int**) (Math.*random*() \* 9 + 1);

**while** (random[0] == random[1]) { // Random Class는 Seed의 변경으로 겹치는 숫자가 나오지 않지만, 메소드는 Seed가 일정하므로

random[1] = (**int**) (Math.*random*() \* 9 + 1);

}

random[2] = (**int**) (Math.*random*() \* 9 + 1);

**while** (random[0] == random[2] || random[1] == random[2]) { // 앞에서 생성된 모든 숫자와 겹치지 않는다면....

random[2] = (**int**) (Math.*random*() \* 9 + 1); // Random Method!

}

**return** random; // 호출에 의해 생성된 Integer type 배열을 리턴한다.

}

MyPanel Panel = **new** MyPanel(); // 결과를 출력하는 Panel 따로 선언

JButton InputBtn = **new** JButton("입력");

Font font = **new** Font("굴림체", Font.***BOLD***, 20); // Font 사용하였음

JPanel Text = **new** JPanel();

JTextField Field = **new** JTextField(15);

**int** Index[] = GenRandom(); // 리턴 배열 저장!, 계속된 호출로 데이터 변조를 방지

**public** Gimal201720970Exam() {

System.***out***.println(Index[0] + " " + Index[1] + " " + Index[2]); // 디버깅을 위한 랜덤 값 출력

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

Container c = getContentPane();

c.setLayout(**new** BorderLayout());

Text.add(Field);

Field.setFont(font);

Field.setEditable(**false**); // 요구점 TextField 수정 불가하도록

Text.add(InputBtn); // panel add..

InputBtn.setFont(font);

InputBtn.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

JButton btn = (JButton) e.getSource();

**if** (btn.equals(InputBtn)) { // 입력된 이벤트가 버튼뿐이지만, 확실히 하기 위함

String num = JOptionPane.*showInputDialog*("3자리 숫자를 입력하세요.");

Field.setText(num);

}

**int**[] input = **new** **int**[3]; // 입력 데이터를 보관하기 위한 배열 선언

String[] temp = Field.getText().split(""); // 각 자리수를 구하기 위해 나눠도 됬지만, 배열로 한번에 받을 수 있는 방법을 선택

**if** (temp.length == 3) { // 입력된 문자의 자릿수가 3자리 일 때만 작동하도록

**try** {

**for** (**int** i = 0; i < temp.length; i++) {

input[i] = Integer.*parseInt*(temp[i]); // String to Integer Casting, But 3자리의 문자열이 입력된다면?

// Error!

}

} **catch** (NumberFormatException exception) {

JOptionPane.*showMessageDialog*(**null**, "3자리 숫자를 입력해주세요.", "오류!", JOptionPane.***ERROR\_MESSAGE***);

}

InputCheck Result = **new** InputCheck(Index, input);

Panel.setData(Result.getStrike(), Result.getBall(), Result.getOut()); // 현재 결과 출력 패널이 분리되어 있으므로,

// 패널에

// 해당 결과를 저장한다.

Panel.repaint(); // 계산된 결과를 기반으로 repaint!

} **else** {

JOptionPane.*showMessageDialog*(**null**, "3자리 숫자를 입력해주세요.", "오류!", JOptionPane.***ERROR\_MESSAGE***); // 잘못된

// 입력임을 인지시킴

}

}

});

c.add(Text, BorderLayout.***NORTH***);

c.add(Panel, BorderLayout.***CENTER***); // 2개의 패널 사용으로 배치 관리자 사용하여 정렬

setTitle("야구 GUI 게임 (201720970 권대한)");

setSize(400, 400);

setVisible(**true**);

}

**public** **static** **void** main(String[] args) {

**new** Gimal201720970Exam();

}

**class** InputCheck {

**int** Strike = 0;

**int** Ball = 0;

**boolean** Out = **false**;

**public** InputCheck(**int**[] random, **int**[] input) { // 입력된 배열을 비교하여 결과를 출력

**for** (**int** i = 0; i < 3; i++) {

**for** (**int** j = 0; j < 3; j++) {

**if** (random[i] == input[j]) {

**if** (i == j) { // 같다면

Strike++;

} **else** { // 다르다면 Ball 임으로...

Ball++;

}

}

}

}

}

**public** **int** getStrike() {

**return** Strike;

}

**public** **int** getBall() {

**return** Ball;

}

**public** **boolean** getOut() {

**if** (Strike == 0 && Ball == 0) {

**return** **true**;

} **else** {

**return** **false**;

}

}

}

**class** MyPanel **extends** JPanel { // paintComponent 사용을 위한 상속 JPanel 사용

Font font = **new** Font("굴림체", Font.***BOLD***, 20);

**int** Strike = 0;

**int** Ball = 0;

**boolean** Out = **false**;

**void** setData(**int** IStrike, **int** IBall, **boolean** IOut) { // 데이터 입력부

Strike = IStrike;

Ball = IBall;

Out = IOut;

}

@Override

**public** **void** paintComponent(Graphics g) { // 결과를 표시하는 Panel에서 setData를 통해 입력받은 결과를 기반으로 도형을 그린다.

**super**.paintComponent(g);

g.setFont(font);

**if** (Strike == 3) {

g.setFont(**new** Font("굴림체", Font.***BOLD***, 50));

g.drawString("홈런입니다.", 60, 160);

} **else** {

**if** (Strike == 0) {

g.setColor(Color.***RED***);

g.drawString("스트라이크", 10, 180);

g.drawRect(120, 160, 25, 25);

g.drawRect(160, 160, 25, 25);

} **else** **if** (Strike == 1) {

g.setColor(Color.***RED***);

g.drawString("스트라이크", 10, 180);

g.fillRect(120, 160, 25, 25);

g.drawRect(160, 160, 25, 25);

} **else** {

g.setColor(Color.***RED***);

g.drawString("스트라이크", 10, 180);

g.fillRect(120, 160, 25, 25);

g.fillRect(160, 160, 25, 25);

}

**if** (Ball == 0) {

g.setColor(Color.***BLUE***);

g.drawString("볼", 90, 220);

g.drawRect(120, 200, 25, 25);

g.drawRect(160, 200, 25, 25);

g.drawRect(200, 200, 25, 25);

} **else** **if** (Ball == 1) {

g.setColor(Color.***BLUE***);

g.drawString("볼", 90, 220);

g.fillRect(120, 200, 25, 25);

g.drawRect(160, 200, 25, 25);

g.drawRect(200, 200, 25, 25);

} **else** **if** (Ball == 2) {

g.setColor(Color.***BLUE***);

g.drawString("볼", 90, 220);

g.fillRect(120, 200, 25, 25);

g.fillRect(160, 200, 25, 25);

g.drawRect(200, 200, 25, 25);

} **else** {

g.setColor(Color.***BLUE***);

g.drawString("볼", 90, 220);

g.fillRect(120, 200, 25, 25);

g.fillRect(160, 200, 25, 25);

g.fillRect(200, 200, 25, 25);

}

**if** (!Out) {

g.setColor(Color.***GREEN***);

g.drawString("아웃", 70, 260);

g.drawRect(120, 240, 25, 25);

} **else** {

g.setColor(Color.***GREEN***);

g.drawString("아웃", 70, 260);

g.fillRect(120, 240, 25, 25);

}

}

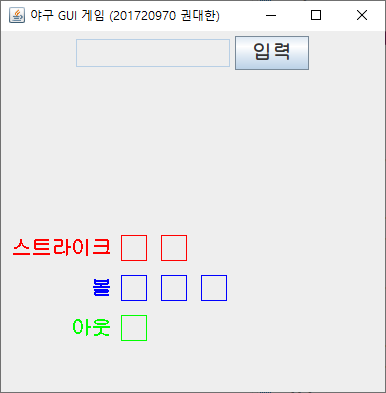
}

}

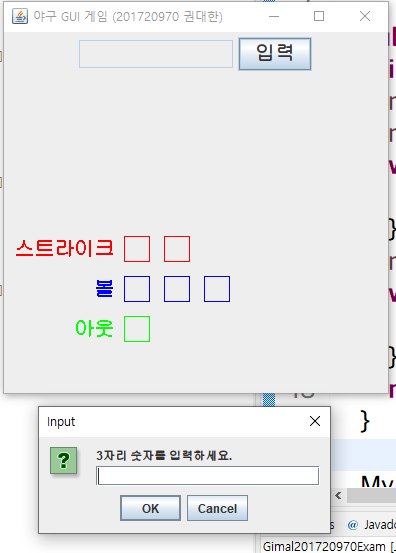
}

[결과 화면]

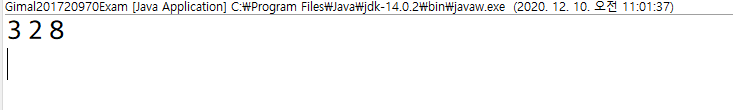
초기 실행 시

****

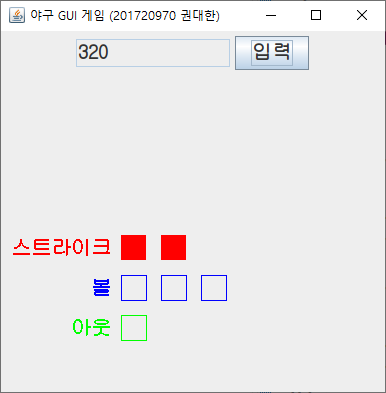
입력 버튼을 눌렀을 때



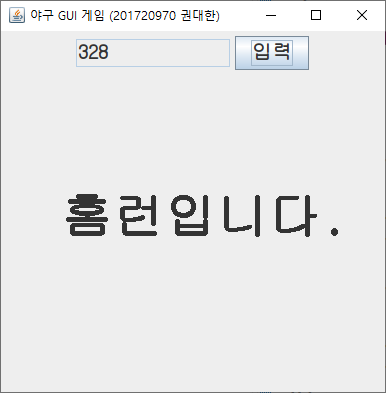
현재 랜덤으로 정해진 숫자



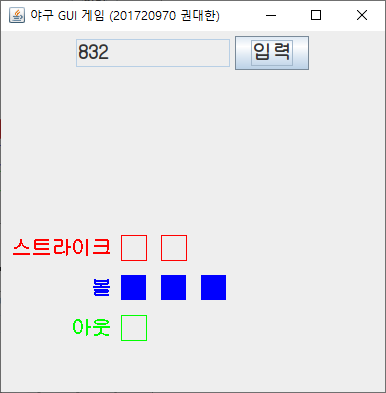
2개만 맞았을 때



전부 맞을 때



3개 다 다르지만, 위치만 다를 때



모든 입력 값의 위치, 숫자 모두 다를 때

