```java
/**
 * header
 */

import java.util.Scanner;
import static java.lang.System.*;
import java.util.Arrays;

public class UltimateRPS
{
    private String playChoice;
    private String compChoice;
    private String[] choices;

    public UltimateRPS()
    {
        playChoice = "";
        compChoice = "";
        choices = new String[] {""};
    } //default constructor

    public UltimateRPS(String player, int choice)
    {
        setPlayers(player, choice);
    } //loaded constructor

    public void setPlayers(String player, int choice)
    {
        //sets the right array to choices depending on the game the player wants to play
        switch (choice)
        {
            case 3:
                choices = new String[] {"rock", "paper", "scissor"};
                break;
            case 5:
                choices = new String[] {"paper", "lizard", "scissor", "rock", "spock"};
                break;
            case 7:
                choices = new String[] {"rock", "water", "air", "paper", "sponge",
                "scissor", "fire"};
                break;
            case 9:
                choices = new String[] {"rock", "gun", "water", "air", "paper",
                "sponge", "human", "scissor", "fire"};
                break;
            case 11:
                choices = new String[] {"rock", "sun", "devil", "water", "air",
                "paper", "sponge", "wolf", "human", "scissor", "fire"};
                break;
            case 15:
                choices = new String[] {"rock", "gun", "lightning", "devil", "dragon",
                "water", "air", "paper", "sponge", "wolf", "tree", "human", "snake",
                "scissor", "fire"};
                break;
            case 25:
                choices = new String[] {"rock", "gun", "dynamite", "nuke", "lightning",
                "devil", "dragon", "alien", "water", "bowl", "air", "moon", "paper",
                "sponge", "wolf", "cockroach", "tree", "man", "woman", "monkey",
                "snake", "axe", "scissor", "fire", "sun"};
            case 101:
                choices = new String[] {"dynamite","helicopter", "tank", "sky", "nuke",
                "laser", "power", "medusa", "lightning", "electricity", "heat",
                "robot", "math", "video game", "fence", "devil", "gold", "platinum",
                "diamond", "dragon", "satan", "mountain", "prayer", "alien", "UFO",
                "rainbow", "TV", "water", "rain", "beer", "cup", "bowl", "guitar",
                "planet", "air", "toilet", "film", "grass", "moon", "airplane",
                "cloud", "paper", "book", "butter", "church", "sponge", "vampire",
                "money", "cross", "community", "brain", "cockroach", "spider", "fish",
                "bird", "cat", "wolf", "duck", "turnip", "tree", "bicycle", "noise",
                "car", "train", "home", "man", "baby", "woman", "police", "princess",
                "prince", "queen", "king", "monkey", "vulture", "porcupine", "blood",
                "snake", "castle", "computer", "peace", "axe", "cage", "poison",
                "scissor", "school", "chainsaw", "fire", "camera", "sun", "wall",
                "death", "rock", "sword", "whip", "law", "gun", "chain", "pit",
                "quicksand", "tornado"};
                break;
            default:
                choices = new String[] {"I", "am", "a", "bad", "coder"};
                break;
        }
        playChoice = player;

        //randomly generates sign for player
        int num = 0 + (int)( Math.random() * choices.length );
        compChoice = choices[num];
    }

    public String determineWinner()
    {
        //if playChoice is the same as compChoice no winner (draw)
        if ( compChoice.equals( playChoice ) )
            return "!Draw Game!";

        //gets index of player's choice
        int index = Arrays.asList(choices).indexOf(playChoice);
        int c = 1;
        for (int i = ( (choices.length) - 1 ) / 2; i > -1; i--)
        {
            //tests to see if it is at the biginning of the array, if it is then moves
            the it to the end of the array
            if (index - c == -1)
            {
                index = choices.length;
                c = 1;
            }
            //tests to see if the string at the point is equal to the computers choice,
            if so then the player wins
            if (compChoice.equals(choices[index - c]))
            {
                return ( "Player Wins <<" + playChoice + " beats " + compChoice + ">>");
            }
            c++;
        }
        return ( "Computer wins <<" + compChoice + " beats " + playChoice + ">>");

        //tried using this but doesnt work for me, wven though its cleaner, I kept
        getting "player wins"
        
        //http://stackoverflow.com/questions/9553058/scalable-solution-for-rock-paper-sci
        ssor
        /*
        switch ( ( ( (choices.length + Arrays.asList(choices).indexOf(playChoice) -
        Arrays.asList(choices).indexOf(compChoice) ) % choices.length ) % 2 ) )
        {
            case 1:
                return "Player Wins <<" + playChoice + " beats " + compChoice + ">>";
            case 0:
                return "Computer wins <<" + compChoice + " beats " + playChoice + ">>";
            default:
                return "";
            }
        */
    }

    public String toString()
    {
        String output="";
        output+="player had " + playChoice+"\n";
        output+="computer had "+ compChoice;
        return output;
    }

    public boolean validWeapon ()
    {
        if (Arrays.asList(choices).indexOf(playChoice) == -1)
            return false;
        else
            return true;
    }
}
```