

KAUN BANEGA CROREPATI PROGRAMME

**ADVANCE JAVA TRAINING**

BY-

Vikas Kashyap R

**INTRODUCTION**

KAUN BANEGA CROREPATI PROGRAMME is well famous and popular game among many players all over the world. This report details the development of a Kaun Banega Crorepati Simulation Game application that is written in Java Programming. In addition, the report details the implementation to solve any kind of Quiz Game. Also, how to generate a Game with different level of difficulties and make sure there will be only one solution. The aim of the report is also to discuss and other logics in order to create and solve Quiz Game. This programme improves methods of solving Quiz .The report concludes by evaluating the end application to analyze how good it met its objectives and the performance of Kaun Banega Crorepati Simulation Game. Finally, the report summaries the overall achievements of the application development and indicates other possible extensions. The main objective of this application is to provide its users with an opportunity to play the famous quizzing game at the comfort of their homes on a computer system. The main purpose of this game is to make people learn new things and improve their general knowledge, just by playing a game. You keep yourself updated by recent current affairs. There is a good scope for improving one’s general knowledge. You can be one step ahead of others. You can be more confident on any topic and gain more knowledge and also able to convey the information to others as you already know about it. By this projects students will Acquire knowledge for developing many more projects like KBC.

# AIMS AND OBJECTIVES

* Promote knowledge and entertainment through an engaging quiz show format.
* Offer a platform for individuals to showcase their knowledge and win prizes.
* Educate and inform audiences about a wide range of topics.
* Inspire learning and curiosity among viewers of all ages.
* Provide a fun and interactive experience for participants and viewers alike.
* Create a sense of community and engagement through a shared interest in trivia and general knowledge.
* Encourage individuals to expand their knowledge and explore new areas of interest.

# GAME RULES

Contestant who is in hot seat will be given a Question with have 4 options. He has to choose the right option from the 4 options.

He has to choose the right option from the 4 options. The Contestant has 3-lifelines i.e. They are:

* 50:50 – The 2 wrong options will be eliminated. The contestant is left with 2 options of which one is the correct option.
* Audience pole: The contestant can ask the audience and choose the right answer
* Skip: If we don’t know the answer we can skip it by choosing the skip option, so that the next question will be displayed without allotting the money for that particular question and this skip option will available for only first five questions.

SOURCE CODE:

import java.util.Scanner; import java.util.ArrayList; import java.util.Collections; import java.util.List;

import java.util.Random;

interface Question { String getQuestion();

String[] getOptions();

String getCorrectAnswer();

}

class MultipleChoiceQuestion implements Question { private String question;

private String[] options; private String correctAnswer;

public MultipleChoiceQuestion(String question, String[] options, String correctAnswer) {

this.question = question; this.options = options;

this.correctAnswer = correctAnswer;

}

@Override

public String getQuestion() { return question;

}

@Override

public String[] getOptions() { return options;

}

@Override

public String getCorrectAnswer() { return correctAnswer;

}

}

public class KBCD {

private static boolean fiftyFiftyUsed = false; private static boolean audiencePollUsed = false; private static boolean skipUsed = false;

public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.println("Enter your name:"); String name = scanner.nextLine();

System.out.println("Welcome to Kaun Banega Crorepati, " + name + "!"); System.out.println("Rules:");

System.out.println(" You will be presented with 10 multiple-choice questions.");

System.out.println(" Each question has 4 options (A, B, C, D)."); System.out.println(" You must select the correct option by entering A, B, C,

or D.");

System.out.println(" If you answer a question incorrectly, the game will end.");

System.out.println(" You can quit the game at any time by typing 'Q' or 'q'."); System.out

.println(" You can use lifelines 'E' (50:50), 'F' (Audience Poll), and 'S' (Skip) for each question.");

System.out.println(" Best of luck!");

System.out.print("Press 'Enter' to start the game..."); scanner.nextLine();

Question[] questions = new Question[10];

questions[0] = new MultipleChoiceQuestion("The first biosphere reserve in India was?",

new String[] { "A.Nanda Devi", "B.Nilgiris", "C.Simlipal", "D.Nokrek"

},

"B");

questions[1] = new MultipleChoiceQuestion(

"Which of these vegetables is actually a modified flower structure?", new String[] { "A.Green Chilly", "B. Tomato", "C.Cauliflower",

"D.Potato" }, "C");

questions[2] = new MultipleChoiceQuestion(

"The wife of which of these famous sports persons was once captain of Indian volleyball team?",

new String[] { "A. K.D.Jadav", "B. Dhyan Chand", "Prakash Padukone", "D. Milkha Singh" }, "D");

questions[3] = new MultipleChoiceQuestion(

"Which wooden religious offering allowed the Greeks to get inside the walls of Troy?",

new String[] { "A. Trojan Dog", "B. Trojan Goat", "C. Trojan Cow", "D. Trojan Hors" }, "D");

questions[4] = new MultipleChoiceQuestion(

"During the company rule, the Hindu Widows Remarriage Act was drafted by ",

new String[] { "A. Lord Canning", "B. Lord Hardinge", "C. Lord Dalhousie", "D. Lord Clive" }, "C");

questions[5] = new MultipleChoiceQuestion(

"What does the ‘HA’ in HAWS, a warfare school run by the Indian Army in Gulmarg stand for?",

new String[] { "A. High Altitude", "B. Heavy Artillery", "C. Hindustan Army", "D. Habitat Adjustme" },

"A");

questions[6] = new MultipleChoiceQuestion("Who is the Author of 'Around The World In Eighty Days'?",

new String[] { "A. Jules Verne", "B.Jonathan Swift", "C. Graham Greene", "D. Herman Melville" }, "A");

questions[7] = new MultipleChoiceQuestion( "Which one is not a tributary of Kaveri River",

new String[] { "A. Hemavati", "B. Bhima", "C. Arkavati", "D. Shimsha"

}, "B");

questions[8] = new MultipleChoiceQuestion( "Who invented Java Programming?",

new String[] { "A. Guido van Rossum", "B. James Gosling", "C.Dennis Ritchie", "D. Bjarne Stroustrup" },

"B");

questions[9] = new MultipleChoiceQuestion( "What is the capital of France?",

new String[] { "A. London", "B. Paris", "C. Berlin", "D. Rome" }, "B");

int[] safeZones = { 4, 7, 9, 10 };

int[] prizeAmounts = { 1000, 2000, 5000, 10000, 20000, 50000, 100000,

250000, 500000, 100000000 };

int prizeMoney = 0;

int currentQuestion = 0;

while (currentQuestion < questions.length) { Question q = questions[currentQuestion];

if (q == null) {

System.out.println("Error: Question not found. Exiting the game."); break;

}

if (skipUsed) { skipUsed = false; currentQuestion++;

continue; // Skip this question

}

System.out.println(q.getQuestion()); for (String option : q.getOptions()) {

System.out.println(option);

}

System.out.print("Choose a lifeline (E - 50:50, F - Audience Poll, S - Skip, N - No lifeline): ");

String lifelineChoice = scanner.next().toUpperCase();

switch (lifelineChoice) { case "E":

if (!fiftyFiftyUsed) { use5050Lifeline(q); fiftyFiftyUsed = true;

} else {

System.out.println("50:50 lifeline has already been used.");

}

break; case "F":

if (!audiencePollUsed) { useAudiencePollLifeline(q); audiencePollUsed = true;

} else {

System.out.println("Audience Poll lifeline has already been used.");

}

break; case "S":

if (currentQuestion < 5 && !skipUsed) { skipUsed = true;

currentQuestion++;

continue; // Skip this question

} else if (currentQuestion >= 5) {

System.out.println("You can't use the 'Skip' lifeline beyond the 5th

question.");

} else {

System.out.println("Skip lifeline has already been used.");

}

break; case "N":

break; default:

System.out.println("Invalid choice. Please choose a valid lifeline.");

}

if (currentQuestion >= 4 && currentQuestion <= 10) { System.out.print("Do you want to quit (Y) or continue (N)? "); String choice = scanner.next().toUpperCase();

if (choice.equals("Y")) {

prizeMoney = prizeAmounts[currentQuestion]; System.out.println("You quit the game. Your total prize money is:

INR " + prizeMoney);

break;

}

}

System.out.print("Your answer (A/B/C/D) or 'Q' to quit: "); String userAnswer = scanner.next().toUpperCase();

if (userAnswer.equals("Q")) {

System.out.println("You quit the game. Your total prize money is: INR "

+ prizeMoney);

break;

}

if (userAnswer.equals("A") || userAnswer.equals("B") || userAnswer.equals("C") || userAnswer.equals("D")) {

if (userAnswer.equals(q.getCorrectAnswer())) { prizeMoney = prizeAmounts[currentQuestion];

System.out.println("Correct answer! You've won: INR " + prizeMoney);

currentQuestion++;

if (currentQuestion == safeZones[0] || currentQuestion == safeZones[1]

|| currentQuestion == safeZones[2] || currentQuestion ==

safeZones[3]) {

System.out.println("Congratulations! You've reached a safe zone.");

}

} else {

System.out.println("Incorrect answer. Game Over!"); break;

quit.");

}

} else {

System.out.println("Invalid choice. Please choose A, B, C, D, or Q to

}

}

System.out.println("Thank you, " + name + " for playing Kaun Banega Crorepati!");

scanner.close();

}

private static void use5050Lifeline(Question question) { System.out.println("Using 50:50 lifeline...");

String[] options = question.getOptions();

int correctOptionIndex = question.getCorrectAnswer().charAt(0) - 'A';

// Generate a list of indexes corresponding to incorrect options List<Integer> incorrectIndexes = new ArrayList<>();

for (int i = 0; i < options.length; i++) { if (i != correctOptionIndex) {

incorrectIndexes.add(i);

}

}

// Randomly select two incorrect indexes Collections.shuffle(incorrectIndexes);

int option1Index = incorrectIndexes.get(0); int option2Index = incorrectIndexes.get(1);

// Remove the incorrect options options[option1Index] = ""; options[option2Index] = "";

// Display the updated options for (String option : options) {

if (!option.isEmpty()) { System.out.println(option);

}

}

}

private static void useAudiencePollLifeline(Question question) { System.out.println("Using Audience Poll lifeline...");

String[] options = question.getOptions();

int correctOptionIndex = question.getCorrectAnswer().charAt(0) - 'A';

Random random = new Random(); int[] percentages = new int[4];

int totalPercentage = 100;

// Assign a higher percentage to the correct option percentages[correctOptionIndex] = random.nextInt(41) + 60; // 60% to 100% totalPercentage -= percentages[correctOptionIndex];

// Distribute the remaining percentage to incorrect options for (int i = 0; i < options.length; i++) {

if (i != correctOptionIndex) {

percentages[i] = random.nextInt(totalPercentage / 2) + 1; // 1% to 50% totalPercentage -= percentages[i];

}

}

// Assign the remaining percentage to the last incorrect option for (int i = 0; i < options.length; i++) {

if (percentages[i] == 0) { percentages[i] = totalPercentage; break;

}

}

// Display the options with accurate percentages for (int i = 0; i < options.length; i++) {

char optionChar = (char) ('A' + i);

System.out.println(optionChar + ". " + options[i] + " - " + percentages[i] + "%");

}

}

}

# UML DIAGRAM