```
1: package dicegames;
2:
3: import java.util.Scanner;
4:
5: public class CrapsPlay {
6:         private Die die1;
7:         private Die die2;
8:         private boolean won;
9:         private int point;
10:        private boolean done;
11:        private Scanner scan;
12:        private boolean sessionDone;
13:        private int wins;
14:        private int losses;
15:
16:        public CrapsPlay() {
17:                die1 = new Die();
18:                die2 = new Die();
19:                won = false;
20:                point = 0;
21:                done = false;
22:                scan = new Scanner(System.in);
23:                sessionDone = false;
24:                wins = 0;
25:                losses = 0;
26:        }
27:
28:        private void welcomeToGame() {
29:                System.out.println("Velkommen til Craps, spillet handler om at rulle rigtigt!");
30:                System.out.println("Starter du med at slå 7 eller 11 vinder du!");
31:                System.out.println("Starter du med at slå 2, 3 eller 12, taber du...");
32:                System.out.println("Slår du alt andet, skal du forsøge at slå det igen, dog uden at slå 7!");
33:                System.out.println("Hvis du stopper med at slå uden at have vundet, taber du også.");
34:        }
35:
36:        private int sum() {
37:                return die1.getFaceValue() + die2.getFaceValue();
38:        }
39:
40:        private void sessionOver() {
41:                System.out.printf("Du har vundet %s gange, og tabt %s gange.\n", wins, losses);
42:                scan.close();
43:        }
44:
45:        private void resetVariables() {
46:                point = 0;
47:                won = false;
48:                done = false;
49:                sessionDone = false;
50:        }
51:
52:        public void startGame() {
53:                welcomeToGame();
54:                while (!sessionDone) {
55:                        resetVariables();
56:                        while (!done) {
57:                                System.out.println("Ønsker du at slå med terningerne? (Ja/Nej)");
58:                                String answer = this.scan.nextLine().toLowerCase();
59:                                if (answer.equals("ja")) {
60:                                        takeTurn();
61:                                } else if (answer.equals("nej")) {
62:                                        done = true;
63:                                } else {
64:                                        System.out.println("Skriv venligst Ja eller Nej som dit svar.");
65:                                }
66:                        }
67:                        gameOver();
68:                        System.out.println("Ønsker du at fortsætte med et nyt spil?(Ja/Nej)");
69:                        String answer = scan.nextLine().toLowerCase();
70:                        if (answer.equals("ja")) {
71:                                sessionDone = false;
72:                        } else if (answer.equals("nej")) {
73:                                sessionDone = true;
74:                        }
75:                }
76:                sessionOver();
77:        }
78:
79:        private void takeTurn() {
80:                this.die1.roll();
81:                this.die2.roll();
82:                System.out.printf("Du slog %s.\n", sum());
83:                if (point == 0) {
84:                        if (sum() == 7 || sum() == 11) {
85:                                won = true;
86:                                done = true;
87:                        } else if (sum() == 2 || sum() == 3 || sum() == 12) {
88:                                won = false;
89:                                done = true;
90:                        } else {
91:                                point = sum();
92:                                done = false;
93:                        }
94:                } else {
95:                        if (sum() == 7) {
96:                                won = false;
97:                                done = true;
98:                        } else if (sum() == point) {
99:                                won = true;
100:                               done = true;
101:                       } else {
102:                               done = false;
103:                       }
104:               }
105:       }
106:
107:       private void gameOver() {
```

```
108:                 if (won) {
109:                         System.out.println("Tillykke du vandt spillet!");
110:                         wins++;
111:                 } else {
112:                         System.out.println("Desværre, du tabte spillet...");
113:                         losses++;
114:                 }
115:         }
116: }
```

```
 1: package dicegames;
 2:
 3: public class CrapsPlayApp {
 4:
 5:         public static void main(String[] args) {
 6:                 CrapsPlay play = new CrapsPlay();
 7:                 play.startGame();
 8:         }
 9:
10: }
```

```java
package dicegames;

/**
 * This class models one Die that can be rolled.
 */
public class Die {
    private int sides;
    private int faceValue;

    /**
     * Constructs a die with six sides.
     */
    public Die() {
        this(6);
    }

    /**
     * Constructs a die with n sides.
     * @param sides the number of sides.
     */
    public Die(int sides) {
        this.sides = sides;
        this.faceValue = 1;
    }

    /**
     * Rolls the die.
     */
    public void roll() {
        faceValue = (int) ((Math.random() * sides) + 1);
    }

    /**
     * Turn the die facing up with a specific face value.
     * @param value the value to face up on the die.
     */
    public void setFaceValue(int value) {
        if (value > 0 && value <= sides) {
            faceValue = value;
        }
    }

    /**
     * Read the die value facing up right now.
     * @return the value of the face
     */
    public int getFaceValue() {
        return faceValue;
    }

}
```

```
1: package dicegames;
2:
3: /**
4:  * This class models one pair of dices. This is useful for games where you have
5:  * to throw two dice at once.
6:  */
7: public class PairOfDices {
8:         /**
9:          * The first die in the pair.
10:         */
11:        private Die die1;
12:        /**
13:         * The second die in the pair.
14:         */
15:        private Die die2;
16:        // Variable to store number of rolls.
17:        private int rolls;
18:        // Variables to store number of facevalues.
19:        private int sixes;
20:        private int fives;
21:        private int fours;
22:        private int threes;
23:        private int twoes;
24:        private int ones;
25:        // Variable to store number of pairs rolled
26:        private int pairs;
27:        // Variable to store highest number rolled
28:        private int high;
29:
30:        /**
31:         * Constructor for objects of class PairOfDices
32:         */
33:        public PairOfDices() {
34:                this.die1 = new Die();
35:                this.die2 = new Die();
36:        }
37:
38:        public PairOfDices(int sides) {
39:                this.die1 = new Die(sides);
40:                this.die2 = new Die(sides);
41:        }
42:
43:        public void resetPairOfDice() {
44:                this.rolls = 0;
45:                this.sixes = 0;
46:                this.fives = 0;
47:                this.fours = 0;
48:                this.threes = 0;
49:                this.twoes = 0;
50:                this.ones = 0;
51:                this.pairs = 0;
52:                this.high = 0;
53:        }
54:
55:        private void testDie(Die die) {
56:                if (die.getFaceValue() == 6) {
57:                        this.sixes++;
58:                } else if (die.getFaceValue() == 5) {
59:                        this.fives++;
60:                } else if (die.getFaceValue() == 4) {
61:                        this.fours++;
62:                } else if (die.getFaceValue() == 3) {
63:                        this.threes++;
64:                } else if (die.getFaceValue() == 2) {
65:                        this.twoes++;
66:                } else if (die.getFaceValue() == 1) {
67:                        this.ones++;
68:                }
69:        }
70:
71:        private void testPair(Die die1, Die die2) {
72:                if (die1.getFaceValue() == die2.getFaceValue()) {
73:                        pairs++;
74:                }
75:        }
76:
77:        public void rollBothDices() {
78:                this.rolls++;
79:                this.die1.roll();
80:                this.die2.roll();
81:                testDie(this.die1);
82:                testDie(this.die2);
83:                testPair(this.die1, this.die2);
84:                if (sumOfDices() > this.high) {
85:                        this.high = sumOfDices();
86:                }
87:        }
88:
89:        public int sumOfDices() {
90:                return this.die1.getFaceValue() + this.die2.getFaceValue();
91:        }
92:
93:        public int getFace1() {
94:                return die1.getFaceValue();
95:        }
96:
97:        public int getFace2() {
98:                return die2.getFaceValue();
99:        }
100:
101:       public int getRolls() {
102:               return rolls;
103:       }
104:
105:       public int getSixes() {
106:               return sixes;
107:       }
```

```
108:
109:        public int getFives() {
110:                return fives;
111:        }
112:
113:        public int getFours() {
114:                return fours;
115:        }
116:
117:        public int getThrees() {
118:                return threes;
119:        }
120:
121:        public int getTwoes() {
122:                return twoes;
123:        }
124:
125:        public int getOnes() {
126:                return ones;
127:        }
128:
129:        public int getPairs() {
130:                return pairs;
131:        }
132:
133:        public int getHigh() {
134:                return high;
135:        }
136:
137: }
```

```
 1: package dicegames;
 2:
 3: import java.util.Scanner;
 4:
 5: public class PigPlay {
 6:         private Die die;
 7:         private Player player1;
 8:         private Player player2;
 9:         private Scanner scan;
10:         private boolean done;
11:         private int winAmount;
12:         private boolean computer;
13:         private int turnRolls;
14:
15:         // The constructor of the PigPlay class.
16:         public PigPlay() {
17:                 die = new Die();
18:                 player1 = new Player();
19:                 player2 = new Player();
20:                 scan = new Scanner(System.in);
21:                 done = false;
22:                 winAmount = 100;
23:                 computer = false;
24:                 turnRolls = 0;
25:         }
26:
27:         // Method used for handling taking a turn for the computer, it is very much the
28:         // same as a human turn, but a seperate method since the logic might be
29:         // drastically different.
30:         private int computerTurn() {
31:                 int result = 0;
32:                 this.turnRolls = 0;
33:                 boolean done = false;
34:                 while (!done) {
35:                         this.turnRolls++;
36:                         die.roll();
37:                         int roll = die.getFaceValue();
38:                         if (roll == 1) {
39:                                 done = true;
40:                                 result = 0;
41:                         } else {
42:                                 result += roll;
43:                                 double randNum = Math.random();
44:                                 if (result / 20 > randNum) {
45:                                         done = true;
46:                                 } else {
47:                                         done = false;
48:                                 }
49:                         }
50:                 }
51:                 return result;
52:         }
53:
54:         // Method to handle when a human player has to take a turn.
55:         private int takeTurn() {
56:                 int result = 0;
57:                 this.turnRolls = 0;
58:                 boolean done = false;
59:                 while (!done) {
60:                         this.turnRolls++;
61:                         die.roll();
62:                         int roll = die.getFaceValue();
63:                         if (roll == 1) {
64:                                 System.out.println("Desværre slog du 1, og får derfor ingen point!");
65:                                 done = true;
66:                                 result = 0;
67:                         } else {
68:                                 result += roll;
69:                                 System.out.printf("Du slog %s, vil du fortsætte(ja) eller få %s point?\n", roll, result);
70:                                 String answer = scan.nextLine().toLowerCase();
71:                                 if (answer.equals("ja")) {
72:                                         done = false;
73:                                 } else {
74:                                         done = true;
75:                                 }
76:                         }
77:                 }
78:                 return result;
79:         }
80:
81:         // A method used for displaying and getting basic information from the players.
82:         // Has a single parameter which is a string to change whether to ask for the
83:         // name of player 2 or the computer.
84:         private void welcomeToGame(String computerHuman) {
85:                 System.out.println("Velkommen til 100.");
86:                 System.out.println("Slå med terning, stop før du slår et, og gem dine point!");
87:                 System.out.println("Hvad er første spillers navn?");
88:                 player1.setName(scan.nextLine());
89:                 System.out.printf("Hvad er %s navn?\n", computerHuman);
90:                 player2.setName(scan.nextLine());
91:                 System.out.printf("Velkommen til 100 %s vs %s\n", player1.getName(), player2.getName());
92:                 System.out.println("Hvor mange point vil i spille til?");
93:                 if (scan.hasNextInt()) {
94:                         winAmount = scan.nextInt();
95:                 }
96:                 System.out.printf("I spiller til %s point.\n", winAmount);
97:         }
98:
99:         // This method initializes the game and is used to ask whether this is a game
100:        // against the computer or another human. The only method except the constructor
101:        // that can be called from outside the class.
102:        public void initialize() {
103:                System.out.println("Vil du spille mod en computer?");
104:                String answer = scan.nextLine().toLowerCase();
105:                if (answer.equals("ja")) {
106:                        computer = true;
107:                } else {
```

```
108:                        computer = false;
109:                }
110:                startGame();
111:        }
112:
113:        // Method for actually starting and running the game.
114:        private void startGame() {
115:                if (computer) {
116:                        welcomeToGame("computerens");
117:                } else {
118:                        welcomeToGame("anden spillers");
119:                }
120:                Player active = player1;
121:                int result = 0;
122:                while (!done) {
123:                        active.addRounds(1);
124:                        if (active.equals(player2) && computer) {
125:                                System.out.println("Det er nu computerens tur, stand by.");
126:                                result = computerTurn();
127:                                active.addRolls(turnRolls);
128:                        } else {
129:                                System.out.printf("Det er nu %s. Tryk på Enter når du er klar.\n", active.getName());
130:                                scan.nextLine();
131:                                result = takeTurn();
132:                                active.addRolls(turnRolls);
133:                        }
134:                        active.addPoints(result);
135:                        System.out.printf("%s fik %s point. %s har nu et total af %s Point.\n", active.getName(), result,
136:                                        active.getName(), active.getPoints());
137:                        if (active.getPoints() >= winAmount) {
138:                                System.out.printf("Tillykke %s du har vundet!\n", active.getName());
139:                                System.out.printf("%s havde i gennemsnit %s slag per runde\n", player1.getName(),
140:                                                player1.averageRolls());
141:                                System.out.printf("%s havde i gennemsnit %s slag per runde\n", player2.getName(),
142:                                                player2.averageRolls());
143:                                done = true;
144:                        } else {
145:                                if (active.equals(player1)) {
146:                                        active = player2;
147:                                } else if (active.equals(player2)) {
148:                                        active = player1;
149:                                }
150:                        }
151:                }
152:        }
153:
154: }
```

```
 1: package dicegames;
 2:
 3: public class PigPlayApp {
 4:
 5:         public static void main(String[] args) {
 6:                 PigPlay play = new PigPlay();
 7:                 play.initialize();
 8:
 9:         }
10:
11: }
```

```java
package dicegames;

public class Player {
	// Class to handle information about a player in the game of pig.
	private String name;
	private int points;
	private int rounds;
	private int rolls;

	public int getRounds() {
		return rounds;
	}

	public void setRounds(int rounds) {
		this.rounds = rounds;
	}

	public void addRounds(int rounds) {
		this.rounds += rounds;
	}

	public int getRolls() {
		return rolls;
	}

	public void setRolls(int rolls) {
		this.rolls = rolls;
	}

	public void addRolls(int rolls) {
		this.rolls += rolls;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public int getPoints() {
		return points;
	}

	public void setPoints(int points) {
		this.points = points;
	}

	public void addPoints(int points) {
		this.points += points;
	}

	public double averageRolls() {
		return rolls / rounds;
	}

}
```

```
 1: package dicegames;
 2:
 3: import java.util.Scanner;
 4:
 5: public class PlayPairOfDice {
 6:
 7:         private Scanner scan;
 8:         private boolean done;
 9:         private PairOfDices dice;
10:
11:         public PlayPairOfDice() {
12:                 this.scan = new Scanner(System.in);
13:                 this.done = false;
14:                 this.dice = new PairOfDices();
15:         }
16:
17:         public void startGame() {
18:                 while (!done) {
19:                         System.out.println("Ønsker du at slå med terningerne? (Ja/Nej)");
20:                         String answer = this.scan.nextLine().toLowerCase();
21:                         if (answer.equals("ja")) {
22:                                 takeTurn();
23:                         } else if (answer.equals("nej")) {
24:                                 this.done = true;
25:                         } else {
26:                                 System.out.println("Skriv venligst Ja eller Nej som dit svar.");
27:                         }
28:                 }
29:                 gameOver();
30:         }
31:
32:         private void takeTurn() {
33:                 this.dice.rollBothDices();
34:                 System.out.printf("Du rullede en sum af %s, med en %s'er og en %s'er.\n", this.dice.sumOfDices(),
35:                                 this.dice.getFace1(), this.dice.getFace2());
36:         }
37:
38:         private void printNumberLine(String num, int value) {
39:                 System.out.printf("Du slog %s %s.\n", value, num);
40:         }
41:
42:         private void gameOver() {
43:                 System.out.printf("Dit største kast var %s.\n", this.dice.getHigh());
44:                 System.out.printf("Du rullede %s par.\n", this.dice.getPairs());
45:                 printNumberLine("enere", this.dice.getOnes());
46:                 printNumberLine("toere", this.dice.getTwoes());
47:                 printNumberLine("treere", this.dice.getThrees());
48:                 printNumberLine("firere", this.dice.getFours());
49:                 printNumberLine("femere", this.dice.getFives());
50:                 printNumberLine("sekesere", this.dice.getSixes());
51:                 this.scan.close();
52:         }
53: }
```

```
 1: package dicegames;
 2:
 3: public class PlayPairOfDiceApp {
 4:
 5:         public static void main(String[] args) {
 6:                 PlayPairOfDice play = new PlayPairOfDice();
 7:                 play.startGame();
 8:         }
 9:
10: }
```

```java
package dicegames;

import java.util.Scanner;

/**
 * A small game where you roll dices.
 * There are no rules; just roll the die until you get bored.
 */
public class PlayRollDie {
    /**
     * How many rolls have been rolled.
     */
    private int rolls;

    /**
     * The scanner used for reading user input.
     */
    private Scanner scan;

    /**
     * The die used in the game.
     */
    private Die die;

    /**
     * Constructs the PlayRollDie game.
     */
    public PlayRollDie() {
        die = new Die();
        scan = new Scanner(System.in);
    }

    /**
     * Print out a neat welcome message to the player.
     */
    private void welcomeToGame() {
        System.out.println("Velkommen til spillet KAST terning");
    }

    /**
     * Finishes the game and prints out the result.
     */
    private void gameOver() {
        System.out.println("Tak for spillet. Du kastede " + rolls + " " + "gange.");
        scan.close();
    }

    /**
     * Take another turn in the game.
     */
    private void takeTurn() {
        die.roll();
        int roll = die.getFaceValue();
        rolls++;
        System.out.println("Du har kastet: " + roll);
    }

    /**
     * Start the game loop.<br/>
     * The game is finished when the player chooses to not roll the die anymore.
     */
    public void startGame() {
        welcomeToGame();

        boolean finished = false;

        while (!finished) {
            System.out.println("Vil du kaste en terning? Angiv Ja eller Nej: ");
            String proceedWithGame = scan.nextLine();
            if (proceedWithGame.equalsIgnoreCase("nej")) {
                finished = true;
            } else {
                takeTurn();
            }
        }

        gameOver();
    }
}
```

```java
package dicegames;

/**
 * This application instantiates the PlayRollDie game and starts it.
 */
public class RollDieApp {

    /**
     * The main method. Don't call this one directly.
     * @param args the program arguments
     */
    public static void main(String[] args) {
        PlayRollDie play = new PlayRollDie();
        play.startGame();
    }

}
```