```
class Casino {
	// Laguna Method:
	// public static void main(String[] args) {
	//	Player player = new Player("Gambler");
	//	int spins = 0;
	//	SlotMachine machine;
	//	// Automatically select the cost of the machine
	//	if (Math.random() < .5) {
	//		machine = new SlotMachine(5);
	//		System.out.println("Welcome, " + player.getName());
	//		System.out.println("You will be using the original machine today, which costs 5 coins");
	//	} else {
	//		machine = new SlotMachine(10);
	//		System.out.println("Welcome, " + player.getName());
	//		System.out.println("You will be using the modded machine today, which costs 10 coins");
	//	}
	//	// Keep playing while the player has enough money
	//	while (player.getMoney() >= machine.getCost()) {
	//		spins++;
	//		machine.pull();
	//		System.out.println(machine);
	//		player.setMoney(player.getMoney() - machine.getCost());
	//		int winnings = machine.calculate();
	//		if (winnings > 0) {
	//			System.out.println("HIT!!! You win " + winnings + " coins!");
	//			player.setMoney(player.getMoney() + winnings);
	//		} else {
	//			System.out.println("Too bad! No win on this pull.");
	//		}
	//		System.out.println(player + "\n\n");
	//	}
	//	System.out.println("GAME OVER! You were able to pull " + spins + " times before going broke!");
	// }

	// My Method:
	public static void main(String[] args) {
		Player player = new Player("Gambler");
		SlotMachine machine = new SlotMachine((Math.random() < 0.5) ? 5 : 10);
		System.out.printf("Welcome, %s%n", player.getName());
		System.out.printf("You will be using the %s machine today, which costs %d coins.%n",
				(machine.getCost() == 10) ? "original" : "modded", machine.getCost());
		int totalSpins = 0;
		for (int spins = 0; player.getMoney() >= machine.getCost(); totalSpins = ++spins) {
			machine.pull();
			System.out.println(machine);
			player.setMoney(player.getMoney() - machine.getCost() + machine.calculate());
			if (machine.calculate() > 0)
				System.out.printf("HIT!!! You win %d coins!%n", machine.calculate());
			else
				System.out.println("Too bad! No win on this pull.");
			System.out.printf("%s%n%n", player);
		}
		System.out.printf("GAME OVER! You were able to pull %d times before going broke",
				totalSpins);
	}

}

class Player {
	private String name;
	private int money;

	Player(String name) {
		this.name = name;
		this.money = 100;
	}

	String getName() {
		return this.name;
	}

	int getMoney() {
		return this.money;
	}

	void setMoney(int money) {
		this.money = money;
	}
```

```
 79
 80         public String toString() {
 81                 return String.format("%s has $%d", this.name, this.money);
 82         }
 83 }
 84
 85 class SlotMachine {
 86         private int cost, multiplier;
 87         private SlotReel sr1, sr2, sr3;
 88
 89         SlotMachine(int cost) {
 90                 this.cost = cost;
 91                 this.multiplier = cost / 5;
 92                 this.sr1 = new SlotReel();
 93                 this.sr2 = new SlotReel();
 94                 this.sr3 = new SlotReel();
 95         }
 96
 97         void pull() {
 98                 this.sr1.spin();
 99                 this.sr2.spin();
100                 this.sr3.spin();
101         }
102
103         int getCost() {
104                 return this.cost;
105         }
106
107         private boolean hit3() {
108                 if (this.sr1.toString().equals("horseshoe")
109                                 || this.sr1.toString().equals("star")
110                                 || this.sr2.toString().equals("horseshoe")
111                                 || this.sr2.toString().equals("star")
112                                 || this.sr3.toString().equals("horseshoe")
113                                 || this.sr3.toString().equals("star"))
114                         return false;
115                 if (this.sr1.equals(sr2) && this.sr2.equals(sr3))
116                         return true;
117                 return false;
118         }
119
120         private boolean hit2() {
121                 if (this.sr1.toString().equals("horseshoe") && this.sr2.toString().equals("horseshoe"))
122                         return true;
123                 return false;
124         }
125
126         int calculate() {
127                 if (hit3()) {
128                         switch (this.sr1.toString()) {
129                                 case "spade":
130                                         return this.multiplier * 20;
131                                 case "diamond":
132                                         return this.multiplier * 30;
133                                 case "heart":
134                                         return this.multiplier * 40;
135                                 case "Liberty Bell":
136                                         return this.multiplier * 50;
137                                 default:
138                                         break;
139                         }
140                 }
141                 if (hit2() && this.sr3.toString().equals("star"))
142                         return this.multiplier * 10;
143                 if (hit2())
144                         return this.multiplier * 5;
145                 return 0;
146         }
147
148         public String toString() {
149                 return String.format("%d: %s|%s|%s", this.cost, this.sr1.toString(), this.sr2.toString(),
150                                 this.sr3.toString());
151         }
152 }
153
154 class SlotReel {
155         private int value;
156
157         void spin() {
158                 this.value = (int) (Math.random() * 6) + 1;
159         }
```

```
160
161         boolean equals(SlotReel reel) {
162                 return this.value == reel.getValue();
163         }
164
165         int getValue() {
166                 return this.value;
167         }
168
169         public String toString() {
170                 switch (this.value) {
171                         case 1:
172                                 return "diamond";
173                         case 2:
174                                 return "heart";
175                         case 3:
176                                 return "spade";
177                         case 4:
178                                 return "horseshoe";
179                         case 5:
180                                 return "star";
181                         case 6:
182                                 return "Liberty Bell";
183                         default:
184                                 return "";
185                 }
186         }
187 }
```