도박의 위험성과 문제를 프로그래밍 세계에서 모델링하고, 이를 해결하는 클래스를 구현하는 것이 목표입니다. 각 클래스를 구성하고, 객체지향 프로그래밍의 개념을 활용하여 도박의 문제점을 표현하는 방식입니다.

**주요 개념**

1. **객체(Object)**: 도박꾼이나 카지노를 하나의 객체로 표현합니다. 이 객체들은 자신의 속성(데이터)과 행동(메서드)을 가지고 있습니다. 예를 들어, 도박꾼 객체는 돈을 베팅하고 잃거나 얻을 수 있는 행동을 할 수 있습니다.
2. **클래스(Class)**: 객체를 만들기 위한 템플릿입니다. 도박꾼을 생성하기 위해 Gambler라는 클래스를 정의하고, 이 클래스에서 도박꾼 객체를 만들 수 있습니다.
3. **속성(Attribute)**: 객체가 가지는 데이터입니다. 도박꾼이라면 이름, 잔액, 중독 수준 등의 데이터를 포함할 수 있습니다.
4. **메서드(Method)**: 객체가 수행하는 행동입니다. 예를 들어, 도박꾼이 돈을 걸고 게임을 하는 메서드를 정의할 수 있습니다.

**클래스 설명**

**1. Gambler (도박꾼 클래스)**

* **역할**: 도박꾼을 모델링한 클래스입니다. 이 클래스는 도박꾼의 행동과 상태를 관리합니다. 도박을 할 때 돈을 걸고 결과에 따라 잔액이 바뀌고, 시간이 지나면서 중독 상태가 증가하는 구조입니다.
* **속성**:
  + name: 도박꾼의 이름을 저장합니다. 도박꾼 객체가 누구인지 구분하기 위한 정보입니다.
  + balance: 도박꾼이 소유한 잔액입니다. 도박에서 이기면 잔액이 증가하고, 지면 잔액이 줄어듭니다.
  + addiction\_level: 도박꾼의 중독 수준을 나타냅니다. 도박을 할 때마다 중독 레벨이 올라갑니다.
  + time\_spent: 도박에 사용한 시간을 기록합니다. 도박을 할 때마다 시간이 지나고, 이를 추적할 수 있습니다.
* **메서드**:
  + gamble(amount): 도박꾼이 일정 금액을 베팅하는 메서드입니다. 주어진 금액을 걸고, 도박 결과에 따라 잔액을 업데이트합니다. 이 메서드에서는 무작위 결과(승리 또는 패배)를 반환하여 도박의 불확실성을 모델링했습니다.
  + check\_balance(): 현재 도박꾼의 잔액을 확인하는 메서드입니다. 도박꾼이 가진 돈이 얼마인지 확인할 수 있습니다.
  + reset\_addiction(): 도박꾼의 중독 수준을 초기화하는 메서드입니다. 중독 레벨이 너무 높아졌을 때, 이를 리셋할 수 있는 방법을 제공합니다.

**2. Casino (카지노 클래스)**

* **역할**: 도박을 제공하는 환경을 모델링합니다. 카지노에서는 도박꾼이 게임을 진행하고, 결과를 처리합니다. 또한 카지노가 얻는 총 이익을 추적할 수 있습니다.
* **속성**:
  + name: 카지노의 이름을 저장합니다.
  + total\_winnings: 카지노가 얻은 총 이익을 나타냅니다. 카지노에서 진행된 도박에서 얼마나 많은 돈을 벌었는지를 기록합니다.
* **메서드**:
  + host\_game(gambler, amount): 도박꾼이 카지노에서 게임을 진행할 때 사용하는 메서드입니다. 도박꾼이 일정 금액을 베팅하고, 결과에 따라 도박꾼의 잔액을 업데이트합니다. 이 메서드를 통해 카지노에서 도박이 실제로 이루어집니다.
  + report(): 카지노의 총 이익을 보고하는 메서드입니다. 카지노가 도박을 통해 얼마나 많은 돈을 벌었는지 확인할 수 있습니다.

**3. AddictionSystem (중독 관리 시스템 클래스)**

* **역할**: 도박 중독 문제를 해결하기 위한 시스템을 모델링합니다. 도박꾼의 중독 수준을 모니터링하고, 위험한 수준에 도달하면 경고를 주고 중독을 리셋하는 역할을 합니다.
* **속성**:
  + max\_addiction\_level: 중독 레벨의 한계를 설정합니다. 도박꾼의 중독 레벨이 이 값을 초과하면 경고를 보내고 중독을 리셋합니다.
* **메서드**:
  + monitor(gambler): 도박꾼의 중독 상태를 감시하는 메서드입니다. 도박꾼의 중독 레벨이 max\_addiction\_level을 초과하면 경고 메시지를 출력하고, 도박꾼의 중독 상태를 리셋합니다.

**코드 실행 흐름**

1. **도박꾼 객체 생성**: Gambler 클래스를 사용해 도박꾼 객체를 생성합니다.

gambler = Gambler(name="홍길동", balance=1000)

1. **카지노 객체 생성**: Casino 클래스를 사용해 카지노 객체를 생성합니다.

casino = Casino(name="Lucky Casino")

1. **중독 시스템 객체 생성**: AddictionSystem 클래스를 사용해 중독 관리 시스템 객체를 생성합니다.

addiction\_system = AddictionSystem()

1. **도박 진행**: 도박꾼이 카지노에서 도박을 진행합니다. 베팅 금액을 걸고, 결과에 따라 잔액이 변화합니다.

casino.host\_game(gambler, 200)

casino.host\_game(gambler, 150)

1. **잔액 확인**: 도박꾼의 잔액을 확인하는 메서드를 호출합니다.

gambler.check\_balance()

1. **중독 상태 모니터링**: AddictionSystem을 사용해 도박꾼의 중독 상태를 감시합니다. 위험한 수준에 도달하면 중독 상태를 리셋합니다.

addiction\_system.monitor(gambler)

**해결 방안**

이 설계에서는 도박 문제를 다루기 위해 세 가지 클래스를 도입했습니다.

* 도박꾼은 돈을 걸고 게임을 진행하며, 중독 레벨이 상승하는 상태를 가집니다.
* 카지노는 도박을 진행하는 환경을 제공하고, 이익을 기록합니다.
* 중독 시스템은 도박꾼의 중독 상태를 모니터링하고, 위험한 수준에 도달하면 경고를 제공하며 중독 상태를 리셋할 수 있습니다.

**이 설계의 의의**

* **중독 관리**: 중독 레벨을 모델링함으로써, 도박의 부작용을 프로그래밍적으로 표현하고, 이를 제어하는 방법을 제시합니다.
* **객체지향적 접근**: 도박과 관련된 요소를 클래스로 분리하여 각자의 역할과 책임을 명확히 하였습니다.