```
Accounts
- username: string
- password: string
- gamesPlayed: int
- gamesWon: int
- best2Normal: int
- best2Streak: int
- best3Odd: int
- best3ThreeKind: int
- best3Power: int
- best3Normal: int
- best4Normal: int
- best4Power: int
- best4Streak: int
- best5Odd: int
- best5Power: int
- best5Streak: int
- best6Normal: int
- best6ThreeKind: int
- best6Power: int
- best6Streak: int
- balance: int
- currGame: MemoryGame()
- unlockedItems: List<shopItem>
+ Accounts(String username, String password)
+ setNewBestScore(int newScore, int dim, int gameMode)
+ setNewStreak(int newStreak, int dim)
+ incrementGamesPlayed()
+ getUsername(): String
+ getPassword(): String
+ get2Normal(): Integer
+ get2Streak(): Integer
+ get3Odd(): Integer
+ get3ThreeKind(): Integer
+ get3Power(): Integer
+ get3Streak(): Integer
+ get4Normal(): Integer
+ get4Power(): Integer
+ get4Streak(): Integer
+ get5Odd(): Integer
+ get5Power(): Integer
+ get5Streak(): Integer
+ get6Normal(): Integer
+ get6ThreeKind(): Integer
+ get6Power(): Integer
+ get6Streak(): Integer
+ getBestScore(int dim, int gameMode): int
+ getBestStreak(int dim): int
+ getGamesPlayed():
+ equals(Object o): boolean
+ setNewGame(MemoryGame game)
+ getCurrGame(): MemoryGame
+ endCurrGame()
+ getBalance(): int
+ setBalance(int balance)
 + getUnlockedItems(): List<shopItem>
+ addUnlockedItem(shopItem item)
+ hasUnlockedItem(shopItem item): boolean
+ incrementBalance(int amount)
+ deductBalance(int amount)
    AccountCollection
     - defAccounts: ArrayList<Accounts
     + AccountCollection()
     + add(Accounts account)
     + getSize(): int
      + getElement(int index): int
      + getScore():
     + getArray(): ArrayList<Accounts>
     + getAccount(String name, String password): Accounts
    + containName(String name): boolean
    + containAccount(Account otherAcct): boolean
```

+ clear()

MemoryGame

- gameMode: int

- numCards: int

- numMatches: int

- isGameActive: int

- starsRevealed: int

lastClickedX: int

- lastClickedY: int

- currStreak: int

- bestStreak: int

+ initGame() + getTheme(): int + getBoard(): Board + gameActive(): boolean

- allowGuiClicks: int

- gameUser: Accounts

+ getNumMoves(): int

+ getNumCards(): int

+ getNumMatches(): int

+ getNumCardsSelected(): int

+ revealColumn(int laserCol)

+ revealAdjacent(int rowBomb, int colBomb)

+ flipCardBackBasic(int maxCardsClicked)

+ updateGamePrompt(String prompt)

+ cardClicked(int row, int col, int guiClick)

+ starPowerReveal(int matchRow, int matchCol)

+ checkMatch(Card card1, Card card2): Boolean

+ checkMatch(Card card1, Card card2, Card card3): Boolean

+ getScore(): int

+ correctGuess()

+ incorrectGuess()

+ getBestStreak(): int

+ updateScore()

+ printBoard()

+ getSize(): int

- gameSubLabel: Label

- gameTheme: int

- timer: Timer

- moves: int

- score: int

- board: Board object

revealedCards: ArrayList<Card>

Board - board: Card [][] size: int gameMode: int cardDesign: int - type1: String[] - type2: String[] + Board(int size) + getSize(): int + initBoard(Accounts user) + fintMatch(int x, int y): int[] + changeMode(int gameMode) + getMode(): int + getCard(int x, int y): Card + shuffle() + toString(): String Card - type1: String - type2: String - revealed: boolean + Card(String type2, String type1) + checkMatch(Card other): Boolean + checkMatch(Card other1 , Card other2) + getType2(): String + getType1(): String + getRevealed(): Boolean + isPower(): int + isStar(): Boolean + isBomb(): Boolean + isLaser(): Boolean

+ MemoryGame(int gameMode, int size, int gameTheme, Accounts user, Label gameSubLabel)

+ MemoryGame(int gameMode, int size, int gameTheme, Accounts user)

```
shopItem
- itemName: String
- price: int
+ shopItem(ItemName: String, price: int)
+ equals(objL Object)
+ getItemName(): String
+ getPrice(): int
shopCollection
- items: ArrayList<shopItem>
+ shopCollection()
+ add(item: shopItem)
+ getSize(): int
+ getElement(index: int): shopItem
+ getArray(): ArrayList<shopItem>
+ clear()
```