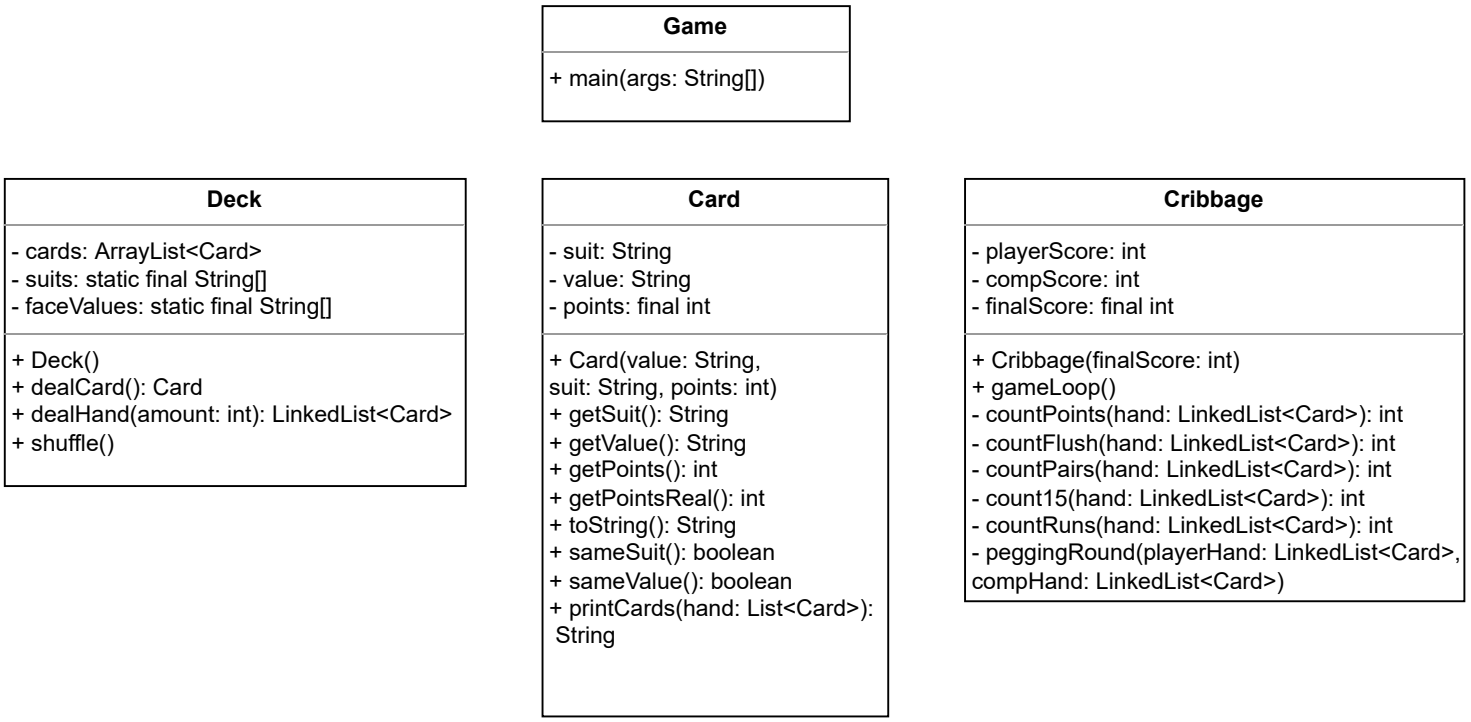
Garrett Martin and Jonathan Homan

Our program uses both Lists and Sets. ArrayList<Card> is used in the Deck class to store the Cards while a TreeSet<Integer> is used in Cribbage method countRuns to check for runs in the hand

One major change that was needed to be made was the addition of an int points variable for each card. This enabled us to both count runs by using a TreeSet easily and also enable the ability check if the cards added up to 15 in any way. This required two get methods, one returns the 1-10 value of cards where 10,J,Q,K overlap and the other returning 1-13 in order to still see the unique value of each card. Another method that was added mainly for convenience was a static method in Card that allowed the ability to easily print out the entire List of Cards in a formatted manner. Other changes involved moving many variables in Cribbage from class wide to inside of methods as well as adding many private methods to allow for easier reading of the gameLoop class and removing the play method as it was no useful. Overall, these changes helped with the readability of the program as well as the ability to add new features more easily since the method structure was more spread out which allowed for adding one way of counting and then testing before adding the next counting method to more easily debug.


Game
+ main(args: String[])
Deck
- cards: ArrayList<Card>
- suits: static final String[]
- faceValues: static final String[]
+ Deck()
+ dealCard(): Card
+ dealHand(amount: int): LinkedList<Card>
+ shuffle()
Card
- suit: String
- value: String
- points: final int
+ Card(value: String,
suit: String, points: int)
+ getSuit(): String
+ getValue(): String
+ getPoints(): int
+ getPointsReal(): int
+ toString(): String
+ sameSuit(): boolean
+ sameValue(): boolean
+ printCards(hand: List<Card>):
String
Cribbage
- playerScore: int
- compScore: int
- finalScore: final int
+ Cribbage(finalScore: int)
+ gameLoop()
- countPoints(hand: LinkedList<Card>): int
- countFlush(hand: LinkedList<Card>): int
- countPairs(hand: LinkedList<Card>): int
- count15(hand: LinkedList<Card>): int
- countRuns(hand: LinkedList<Card>): int
- peggingRound(playerHand: LinkedList<Card>,
compHand: LinkedList<Card>)