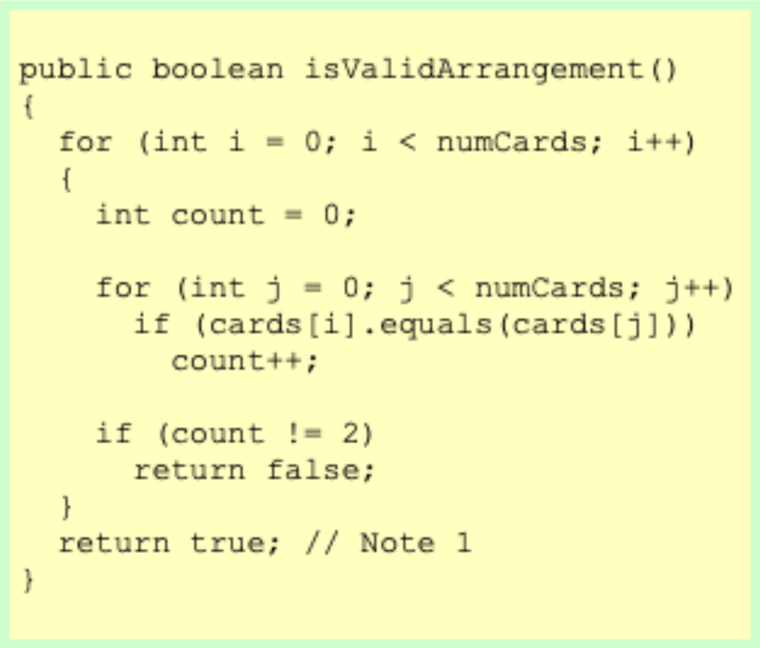
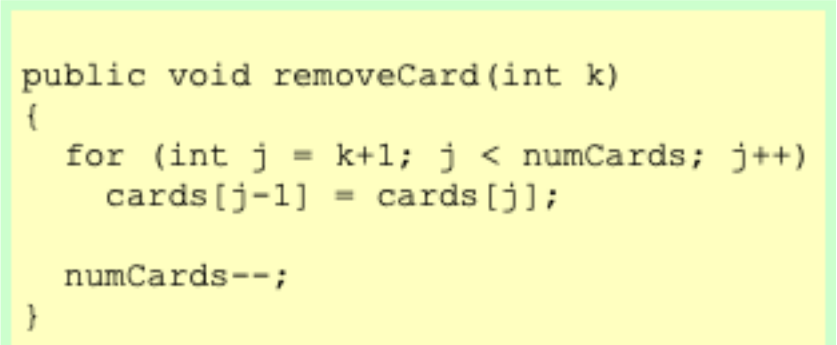
MemoryLane solution

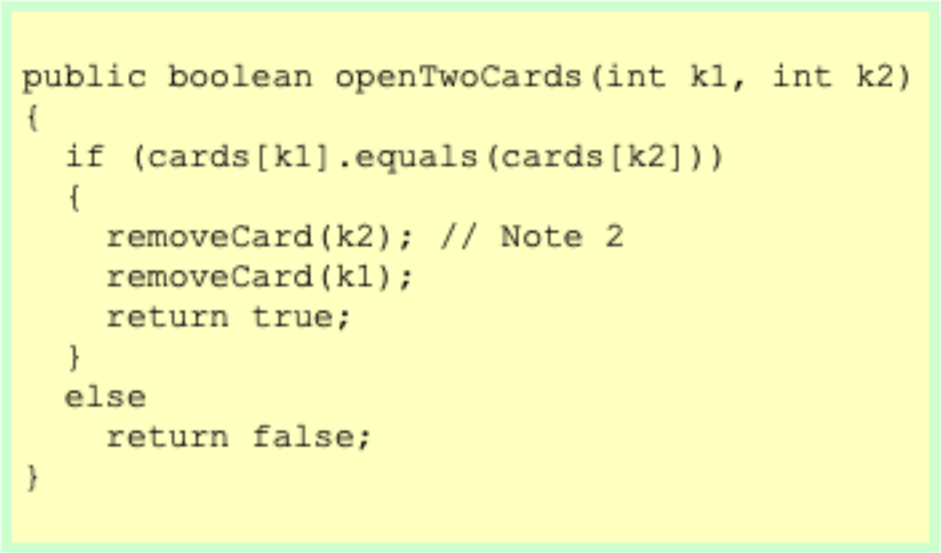


```
public boolean isValidArrangement()
{
  for (int i = 0; i < numCards; i++)
  {
    int count = 0;

    for (int j = 0; j < numCards; j++)
      if (cards[i].equals(cards[j]))
        count++;

    if (count != 2)
      return false;
  }
  return true; // Note 1
}
```

Note 1: `return true` only outside of the outer `for` loop.



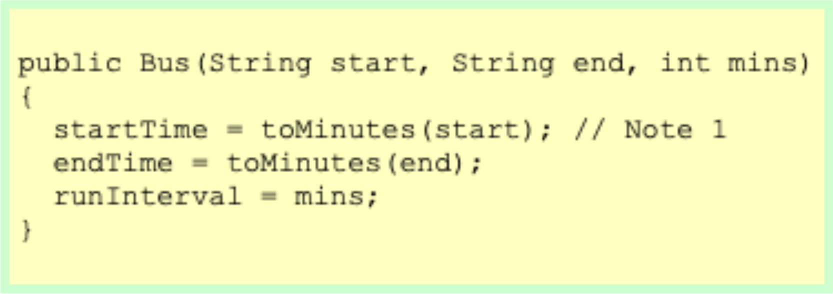
```
public void removeCard(int k)
{
  for (int j = k+1; j < numCards; j++)
    cards[j-1] = cards[j];

  numCards--;
}
```



```
public boolean openTwoCards(int k1, int k2)
{
  if (cards[k1].equals(cards[k2]))
  {
    removeCard(k2); // Note 2
    removeCard(k1);
    return true;
  }
  else
    return false;
}
```

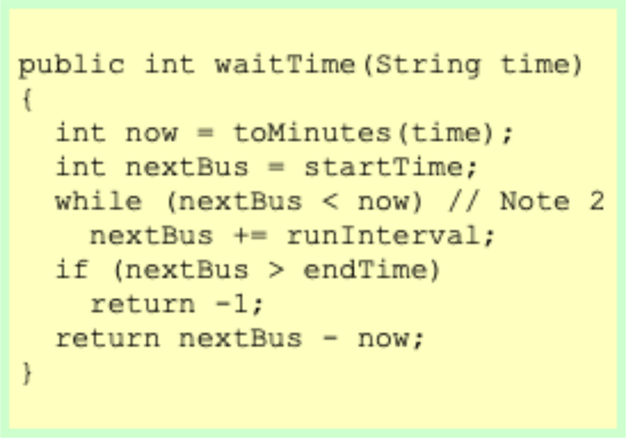
Note 2: It is important to remove the cards in the correct order, because when a card is removed, the subsequent cards shift to the left.

Bus Solution



```
public Bus(String start, String end, int mins)
{
  startTime = toMinutes(start); // Note 1
  endTime = toMinutes(end);
  runInterval = mins;
}
```

Note 1: It is OK to call public or private instance methods of a class from its constructors.



```
public int waitTime(String time)
{
  int now = toMinutes(time);
  int nextBus = startTime;
  while (nextBus < now) // Note 2
    nextBus += runInterval;
  if (nextBus > endTime)
    return -1;
  return nextBus - now;
}
```

Note 2: Of course it is possible to calculate the expected next bus arrival time using arithmetic, but it is much easier to use a loop.