

Arrays Revisited

Arrays of object references



1

Arrays of Objects

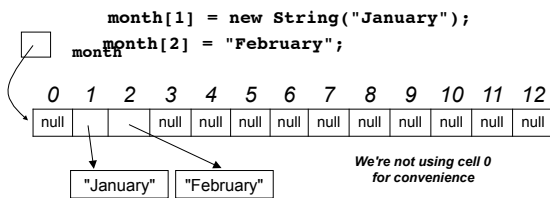
- We can use arrays to hold a collection of references to objects of the same type.
 - Technically in Java, the types of the objects do not have to be exactly the same, but for now, we'll assume they are.
- Initially, when we declare an array of object references, all cells of the array contain the value **null**.
- Example:

```
String[] month = new String[13];
```

2

Arrays of Objects

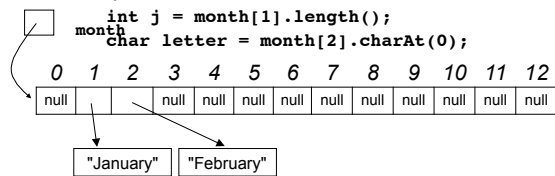
- Each cell of an array of object references can hold one reference to an object.
- Example:



3

Arrays of Objects

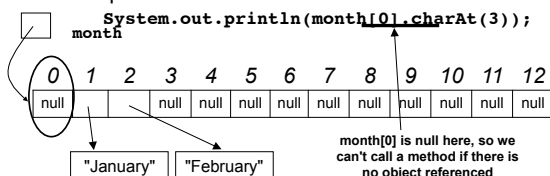
- Using the array name and subscript, we obtain a reference. If the reference is not null, we can call a method on the object that is referenced.
- Example:



4

NullPointerException

- Using the array name and subscript, we obtain a reference. If the reference is null, we cannot call a method on the object that is referenced.
- Example:



5

Example

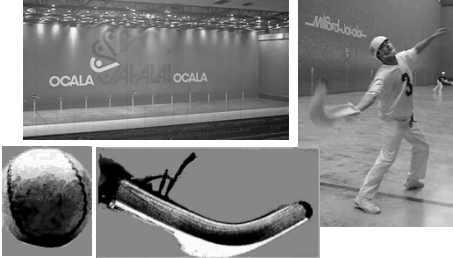
- Assume that all 12 month names are referenced from our array **month**.
 - Remember: we're not using cell 0 this time.
- To print out all of the month names in the format:

```
Jan
Feb
Mar
...etc...
```

```
for (int i = 1; i <= 12; i++)
    System.out.println(
        month[i].substring(0,3));
```

6

Jai Alai



7

Official Jai Alai Rules

- Usually 8 teams participate. Teams line up in order 1,2,3,4,5,6,7,8.
- Team 1 plays team 2. The winner earns 1 point and stays on the court to play the next team in line; the loser goes to the end of the line.
- After all teams have played once, point values double for each match.
- The first team to reach 7 points (sometimes 9) is the winner of the game.

8

A class to model a Jai Alai Team

```
public class Team {
    private int teamNumber;
    private int teamScore;
    public Team(int number) {
        teamNumber = number;
        teamScore = 0;
    }
    public int getTeamNumber() {
        return teamNumber;
    }
    public int getTeamScore() {
        return teamScore;
    }
}
```

9

A class to model a Jai Alai Team (cont'd)

```
public void addPoints(int numPoints) {
    if (numPoints >= 1) {
        teamScore += numPoints;
    }
}
public String toString() {
    return ("Team " + teamNumber + ": "
        + teamScore);
}
```

10

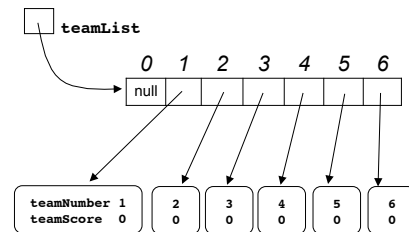
Jai Alai: Initializing the array

```
final int NUM_TEAMS = 6; We'll use just 6 teams here for now.
Team[] teamList
    = new Team[NUM_TEAMS+1];
for (int position = 1; We're not using cell 0 for convenience again
    position <= NUM_TEAMS; position++)
{
    teamList[position] = new Team(position);
}
```

create a new Team with a number equal to its position in the array

11

Jai Alai: Initializing the array



12

Jai Alai: Playing a match



- The teams in the first two positions (positions 1 & 2) play each other.
 - Note: This isn't necessarily Teams 1 & 2, except at the beginning of the Jai Alai game.
- We wish to store the winner of the match in position 1 and the loser in position 2 of the array.

13

Jai Alai: Moving the winner into position 1



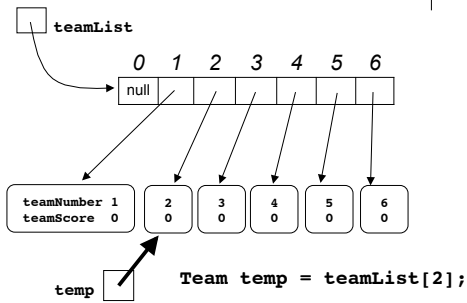
- If the winner was the team in position 1, we have no work to do.
- If the winner is the team in position 2:

```
Team temp = teamList[2];
teamList[2] = teamList[1];
teamList[1] = temp;
```

This is called a SWAP.

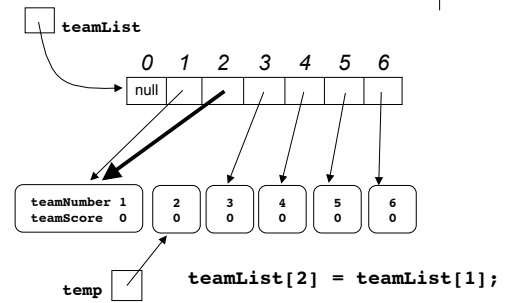
14

Jai Alai: Swapping two teams



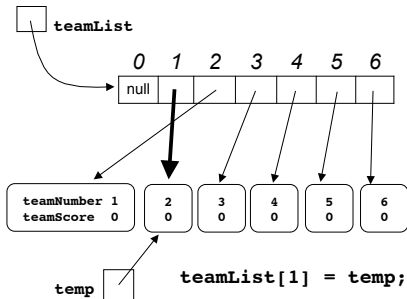
15

Jai Alai: Swapping two teams



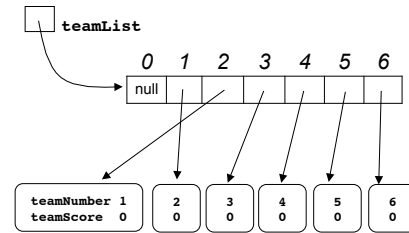
16

Jai Alai: Swapping two teams



17

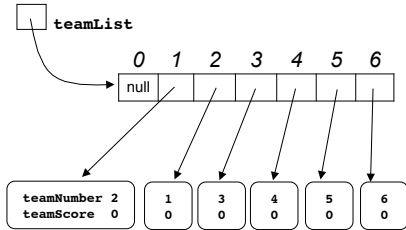
Jai Alai: Swapping two teams



Actually, the objects never move;
it's the references that change...

18

Jai Alai: Swapping two teams



...but we can untwist the references to make it easier to see.

19

Jai Alai: Moving the loser to the end

- The loser (who must be in position 2 now) must move to the end of the line.
- We must shift all other teams "forward" one position (toward the "beginning" of the array) and then reinsert the loser in the last cell of the array.

20

Jai Alai: Moving the loser to the end

```

loser = teamList[2];
// shift waiting teams to the left one position

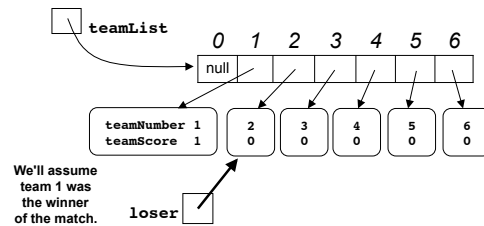
for (int position = 3;
     position <= NUM_TEAMS; position++)
{
    teamList[position-1] = teamList[position];
}

// insert loser of match at end of waiting line
teamList[NUM_TEAMS] = loser;
    
```

21

Jai Alai: Moving the loser to the end

```
loser = teamList[2];
```

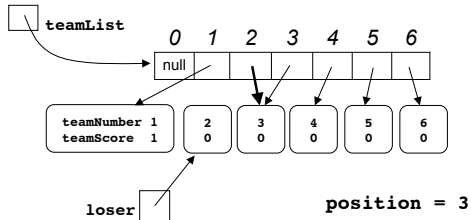


22

Jai Alai: Moving the loser to the end

```

for (int position = 3; position <= NUM_TEAMS; position++)
    teamList[position-1] = teamList[position];
    
```

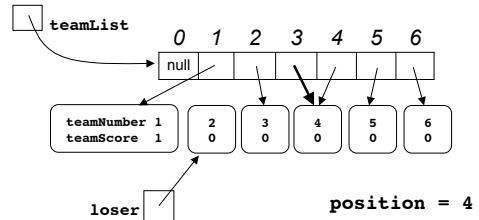


23

Jai Alai: Moving the loser to the end

```

for (int position = 3; position <= NUM_TEAMS; position++)
    teamList[position-1] = teamList[position];
    
```

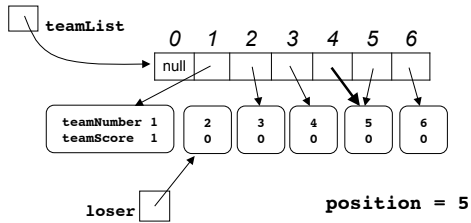


24

Jai Alai: Moving the loser to the end



```
for (int position = 3; position <= NUM_TEAMS; position++)
    teamList[position-1] = teamList[position];
```

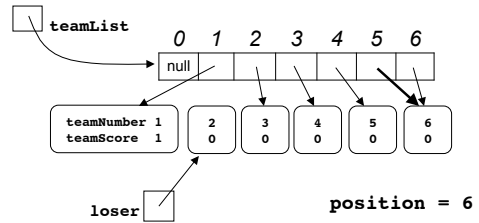


25

Jai Alai: Moving the loser to the end



```
for (int position = 3; position <= NUM_TEAMS; position++)
    teamList[position-1] = teamList[position];
```

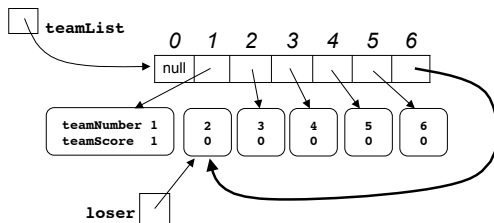


26

Jai Alai: Moving the loser to the end



```
teamList[NUM_TEAMS] = loser;
```

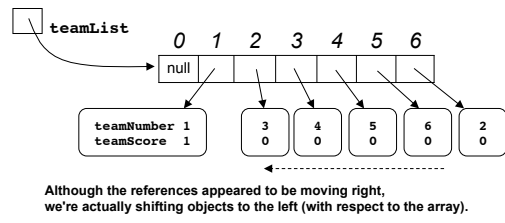


27

Jai Alai: Moving the loser to the end



OR (untwist the references):



28

Shifting data in an array



- If you want to shift data values to the "left" one position (toward the beginning of the array):
 - Work from left to right.
 - Copy from each position to position-1.

```
for (int position = 3; position <= 6; position++)
    teamList[position-1] = teamList[position];
    to           from
```

- Question: What should you do if you want to shift data to the "right" one position (toward the end of the array)?

29

What's next...



- Given an array of references to objects:
 - Insert a new object reference into the array at various positions based on some criteria.
 - Remove the reference of an object from the array given some criteria.
 - Count the number of objects referenced in the array that match some criteria.
 - Create a new array with object references from the original array with only those objects that meet some criteria.

30