


# Classes: Part 2



Memory

Diagram





```
public static void main(String[] args) {  
    Player p1 = new Player("Dark Cecil", 10);  
    Player p2 = new Player("Kain", 14);  
    Player p3 = p1;  
    p1.setName("Paladin");  
    System.out.println(p3.getName());  
}
```

Stack		Heap
Name	Value	
		<u>in/out</u>

- We'll trace this version of the code
- Set up the stack, heap, and in/out



```

➡ public Player(String name, int maxHP) {
    this.setMaxHP(maxHP);
    this.setHP(maxHP);
    this.setName(name);
}

public void setMaxHP(int maxHP) {
    this.maxHP = maxHP;
}

public void setHP(int hp) {
    if (hp <= this.maxHP) {
        this.hp = hp;
    } else {
        this.hp = this.maxHP;
    }
}

public String getName() {
    return this.name;
}

public void setName(String name) {
    this.name = name;
}

➡ public static void main(String[] args) {
    Player p1 = new Player("Dark Cecil", 10);
    Player p2 = new Player("Kain", 14);
    Player p3 = p1;
    p1.setName("Paladin");
    System.out.println(p3.getName());
}

```

Stack		Heap
Name	Value	
		<div style="border: 1px solid black; width: 60%; height: 80px; margin: 10px auto;"></div> <p style="text-align: center;"><u>in/out</u></p> <div style="border: 1px solid black; padding: 10px; margin-top: 20px;"><ul style="list-style-type: none"><li>New objects are created on the heap</li><li>Only a reference is stored in variables</li></ul></div>



```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    ➡ public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

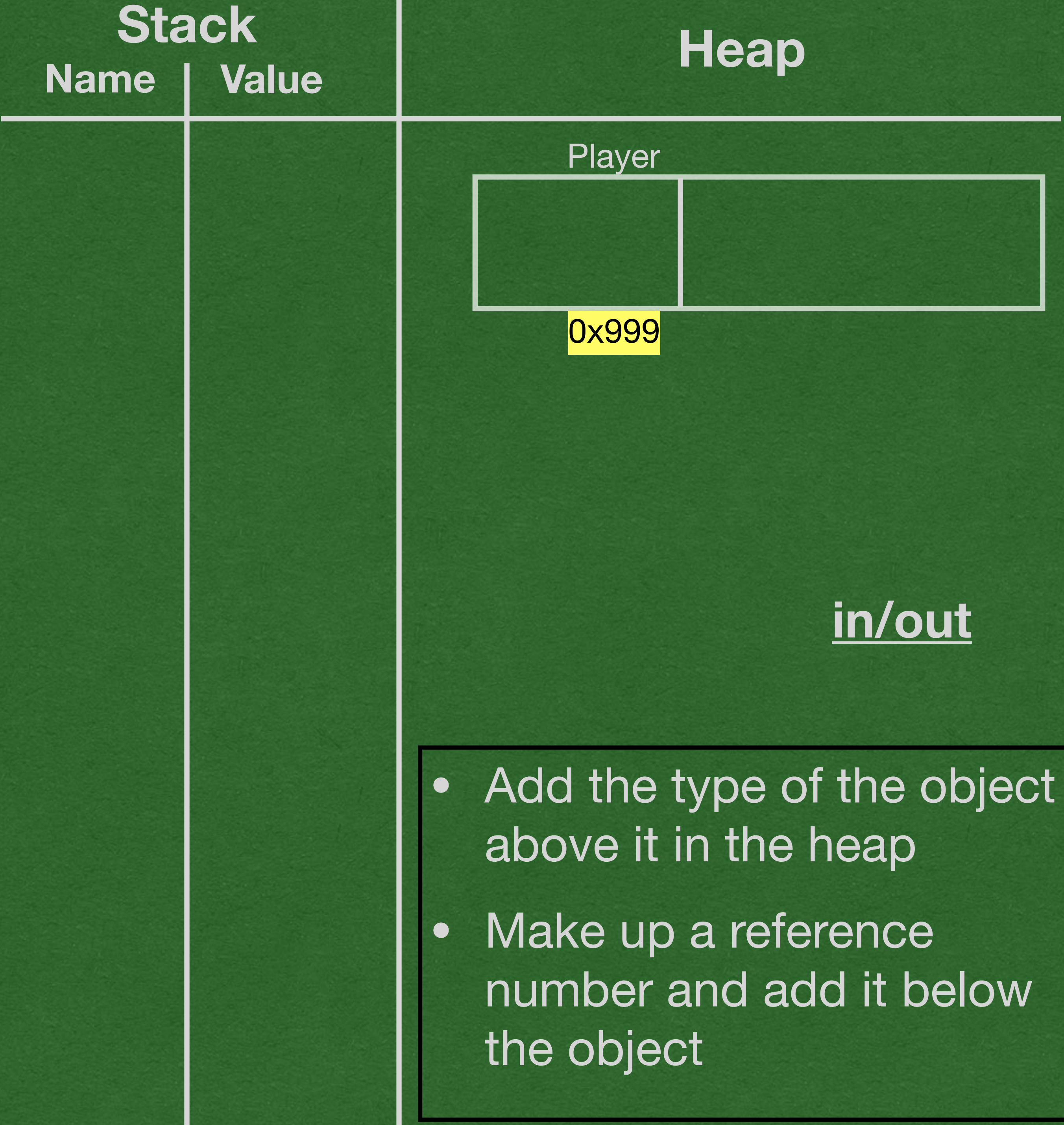
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    ➡ public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    ➡ public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

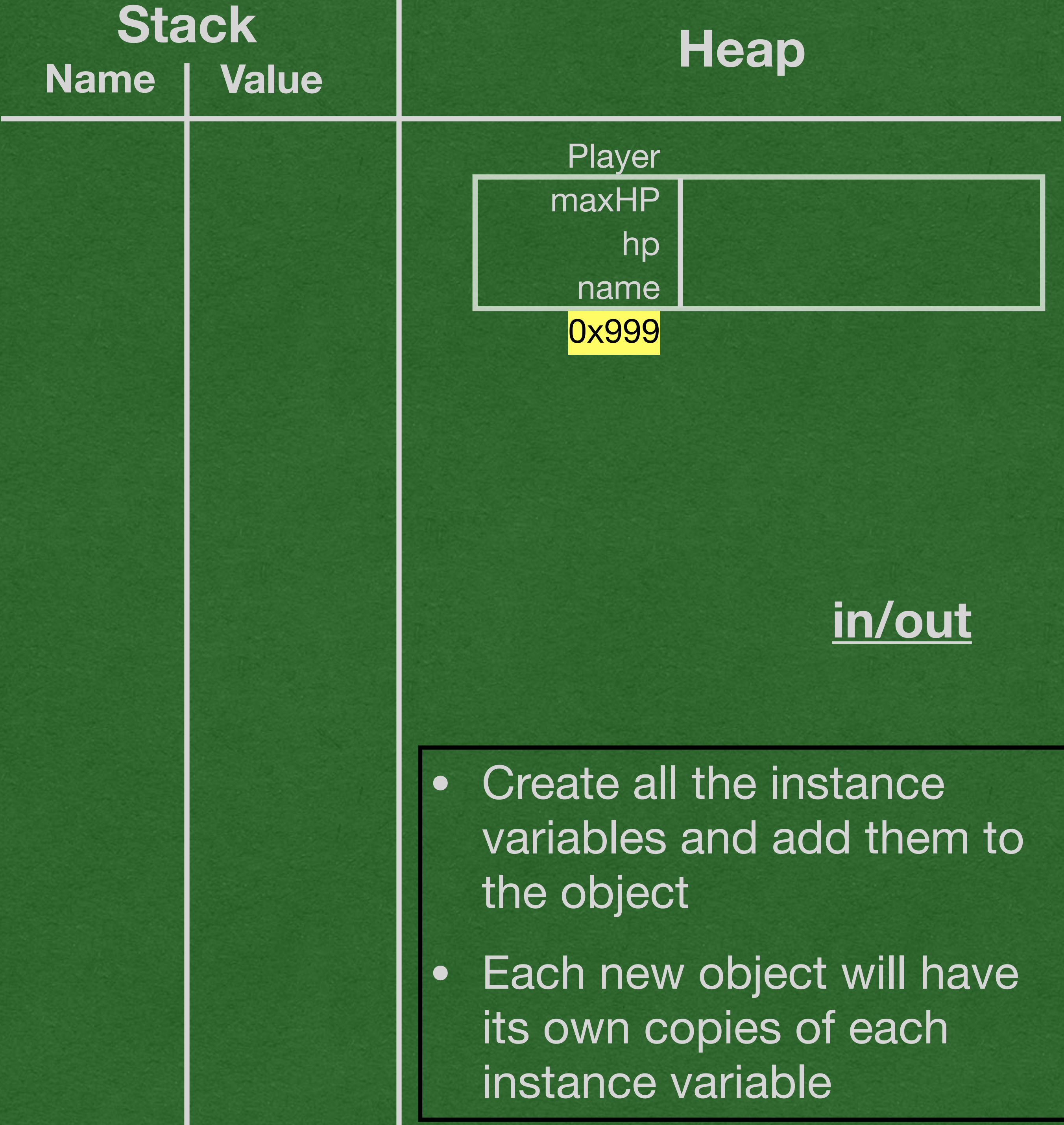
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    ➡ public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    ➡ public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

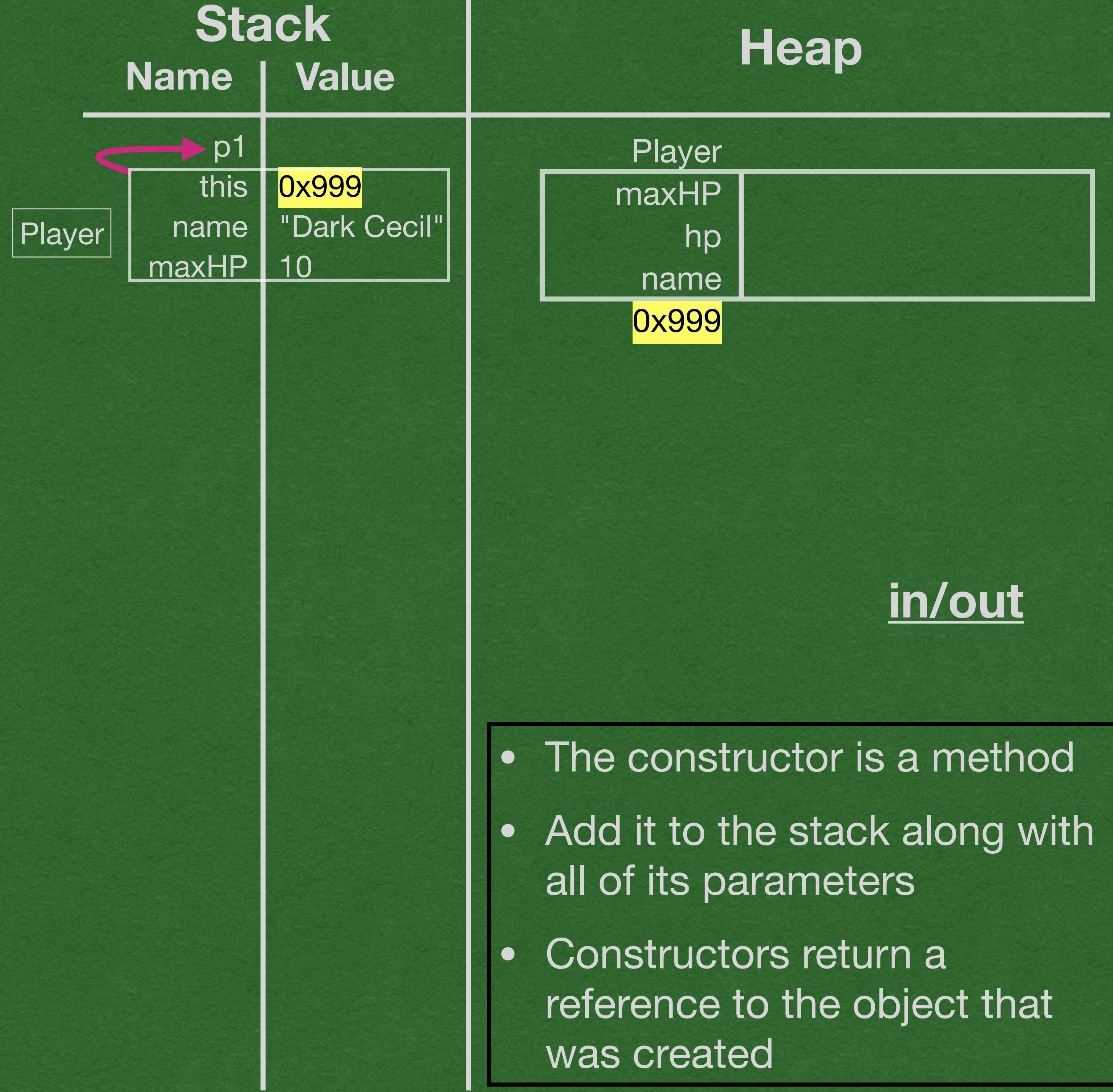
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    ➡ public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    ➡ public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

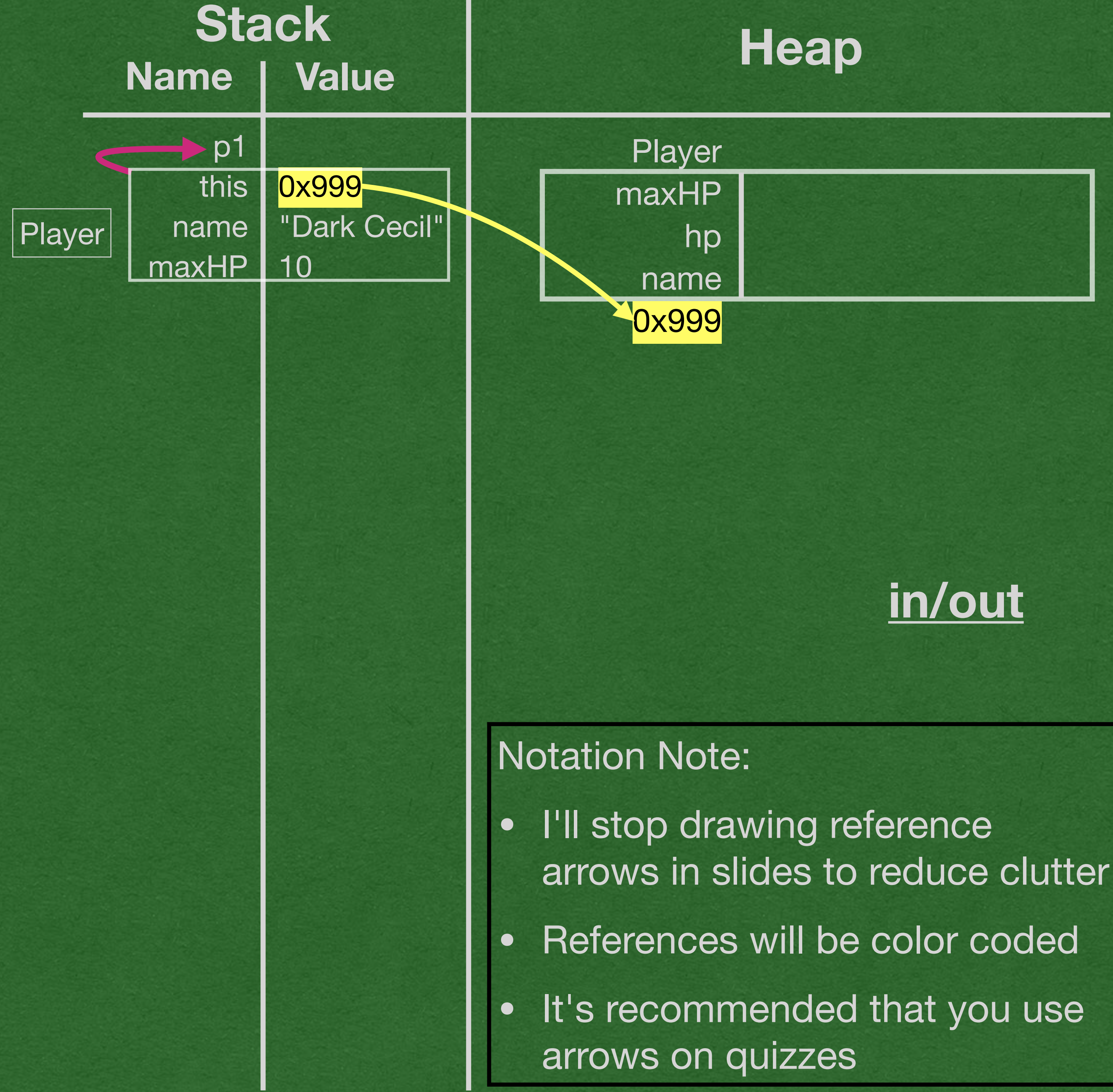
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    ➡ public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    ➡ public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

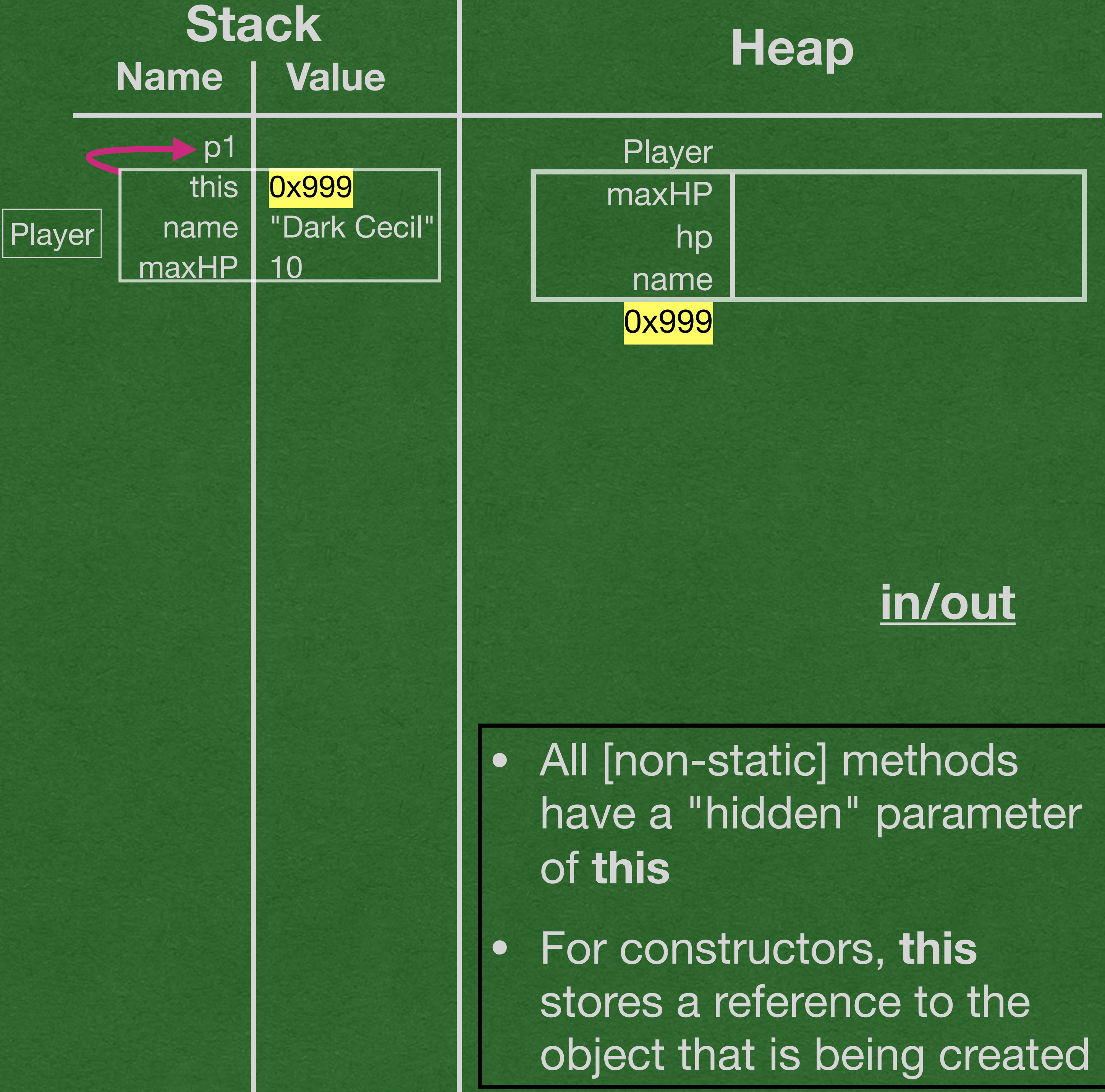
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    ➡ public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

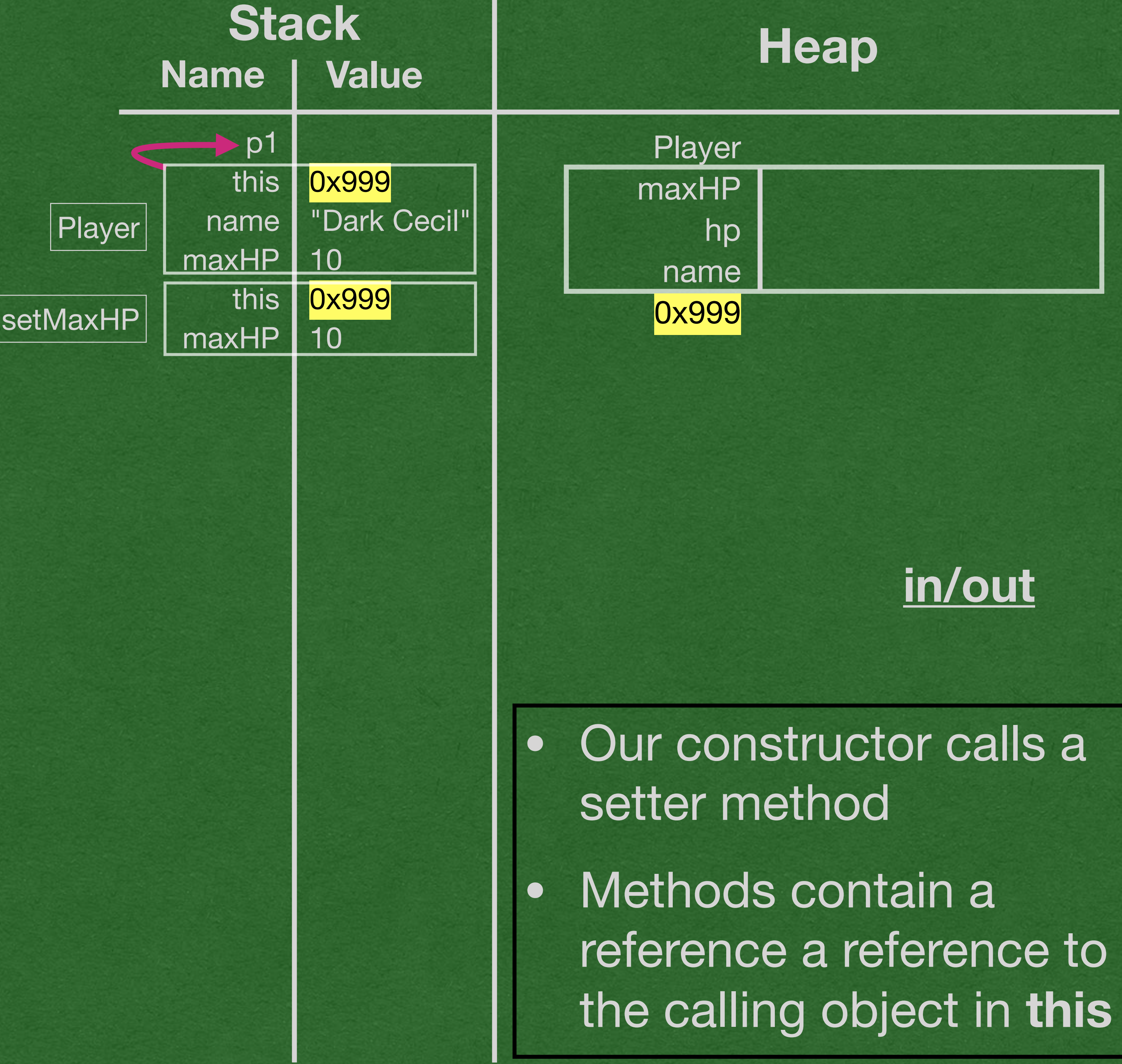
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

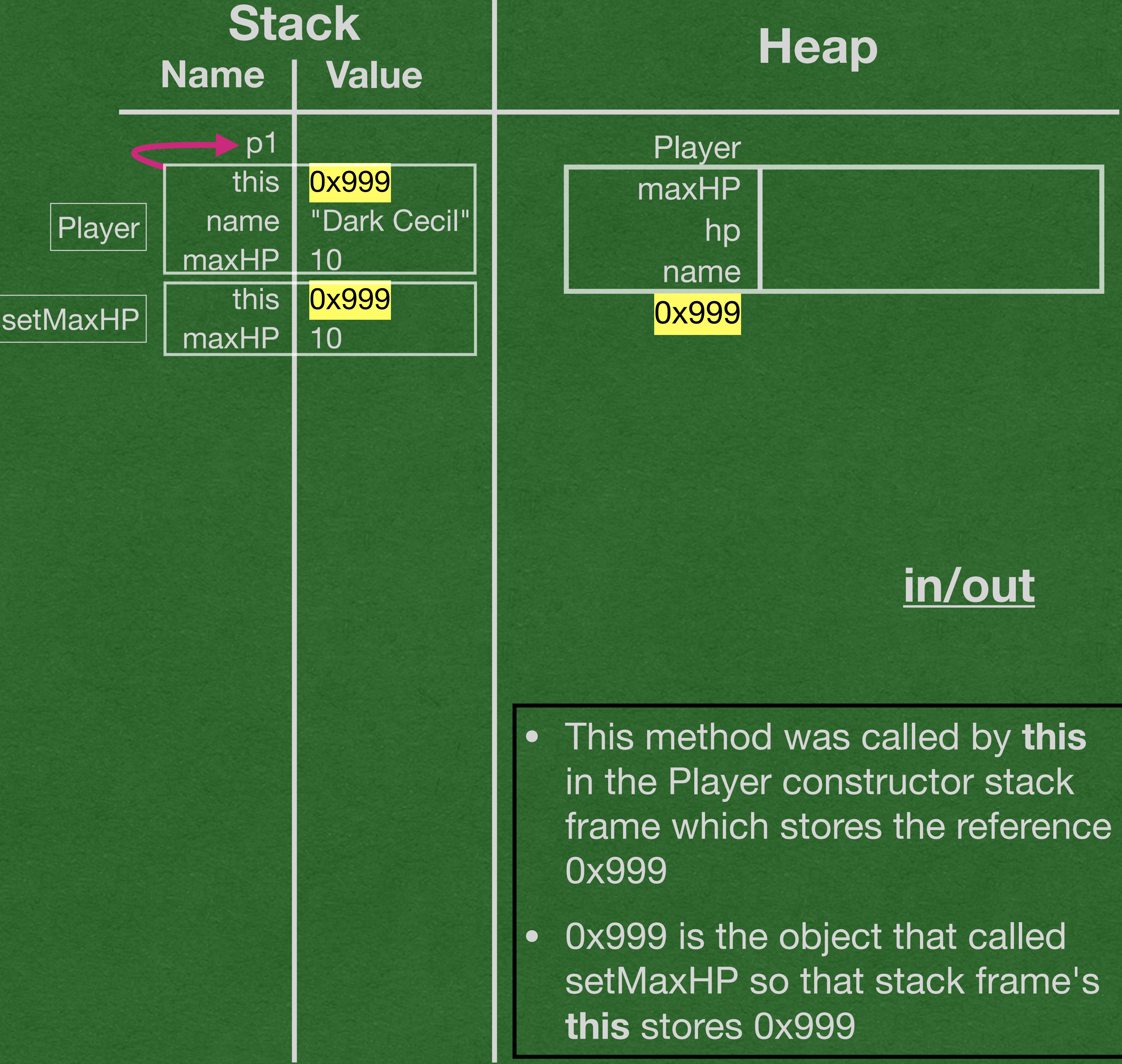
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    ➡ public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

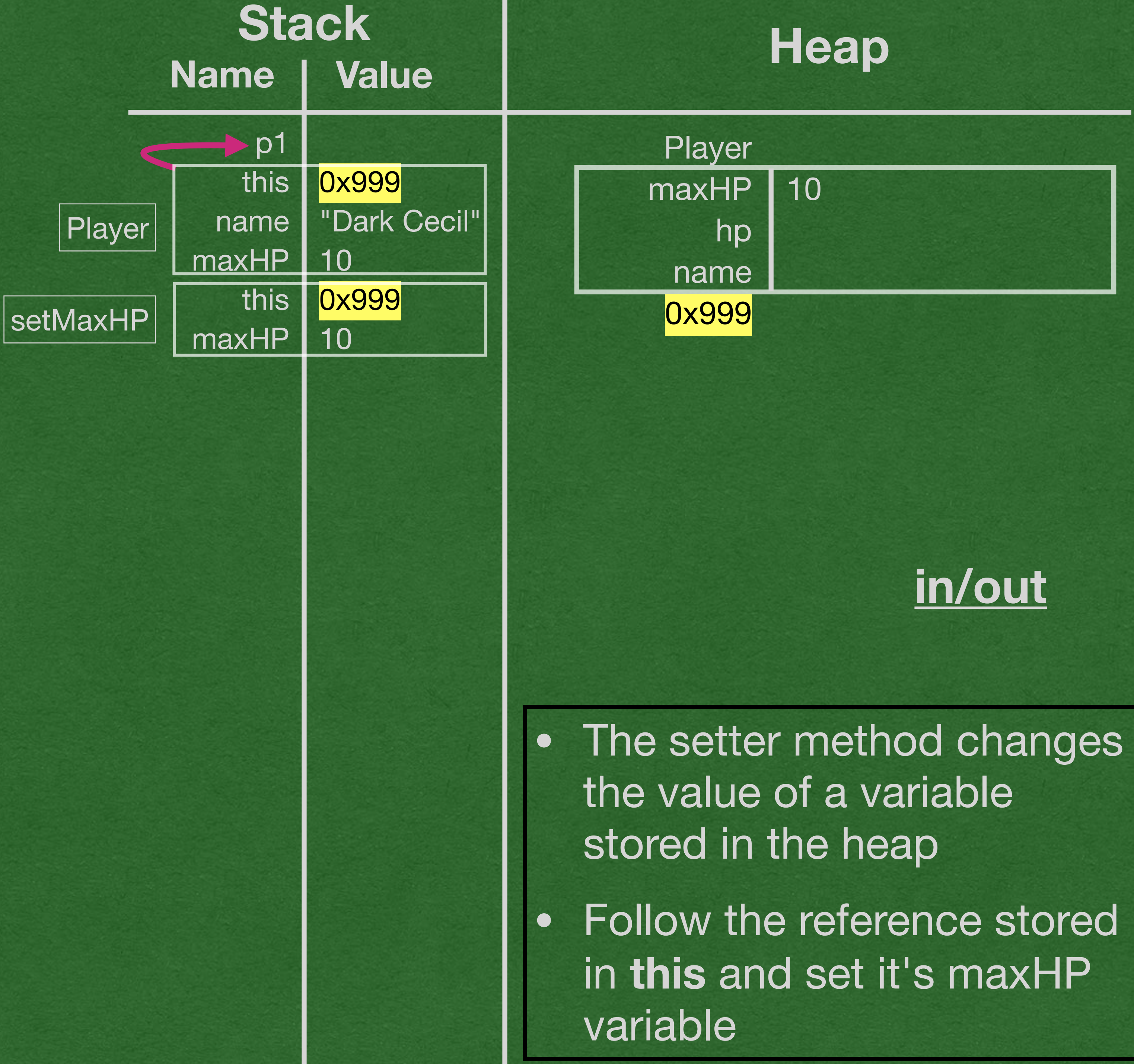
    ➡ public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    ➡ public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```

Stack	
Name	Value
Player	p1
	this 0x999
	name "Dark Cecil"
setMaxHP	maxHP 10
	this 0x999
	maxHP 10

Heap

Player	
maxHP	10
hp	
name	

0x999

in/out

Notation Note:

- I'll gray out a stack frame that is removed from the stack
- This will have the same meaning as crossing it out
- Makes the variables readable



```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```

Stack	
Name	Value
Player	p1
	this 0x999
	name "Dark Cecil"
setMaxHP	this 0x999
	maxHP 10
setHP	this 0x999
	hp 10

Heap

Player	
maxHP	10
hp	10
name	

0x999

in/out

- Calling setHP will set the hp variable on the stack for this object



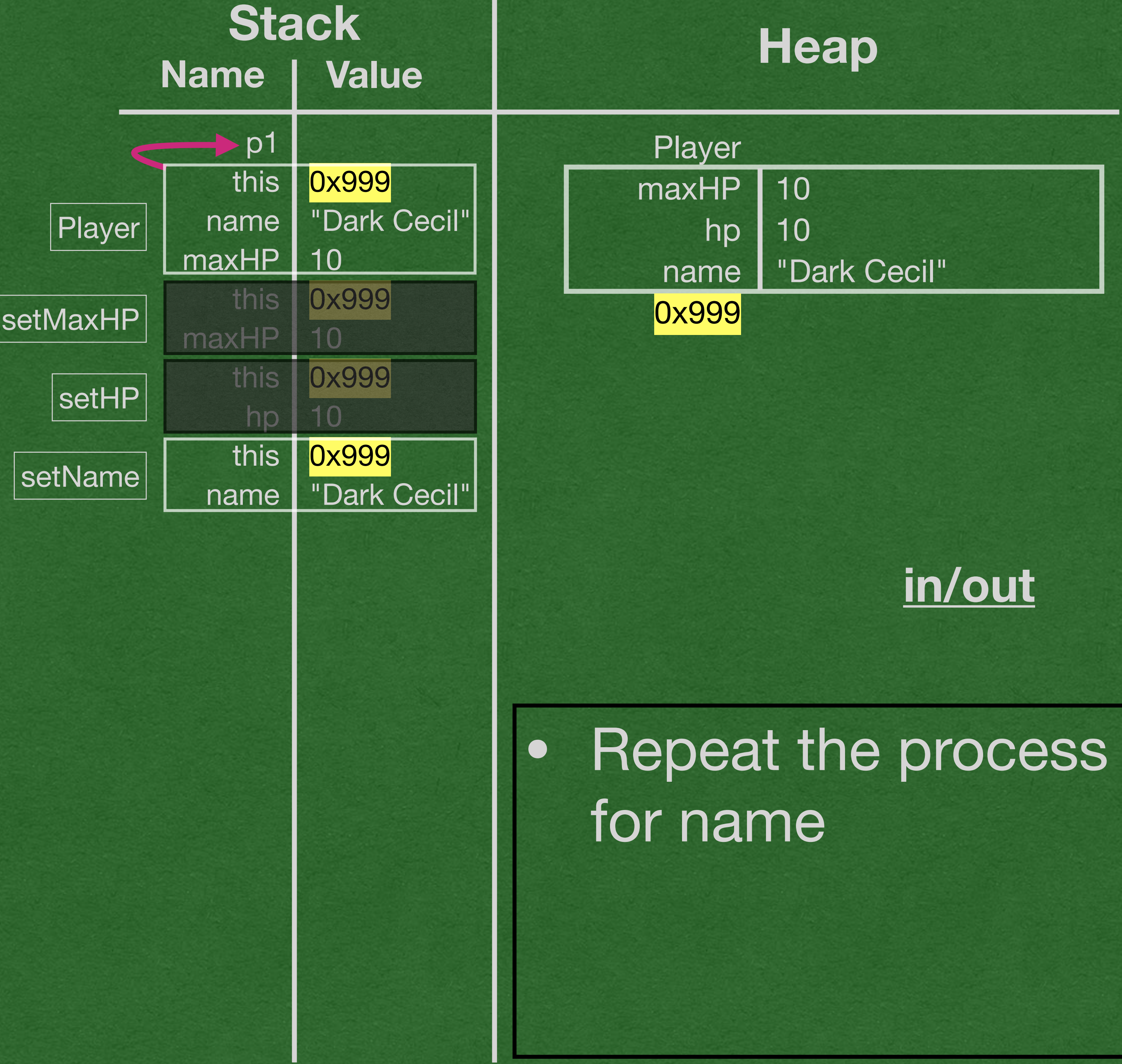
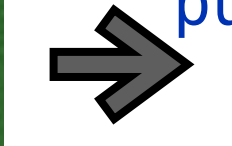
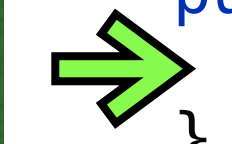
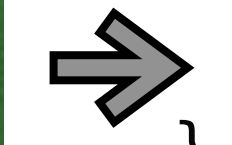
```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }
    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }
    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

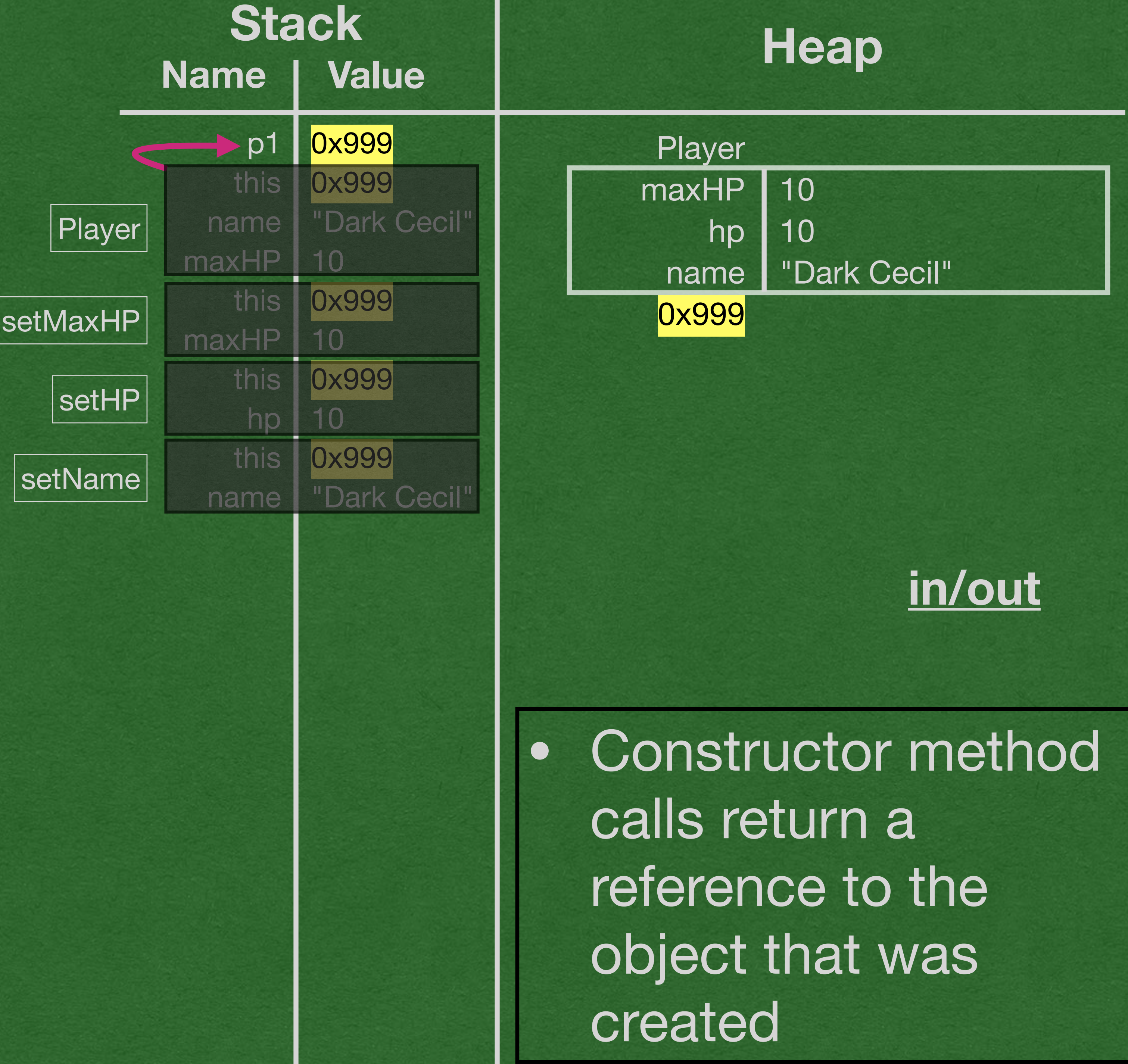
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

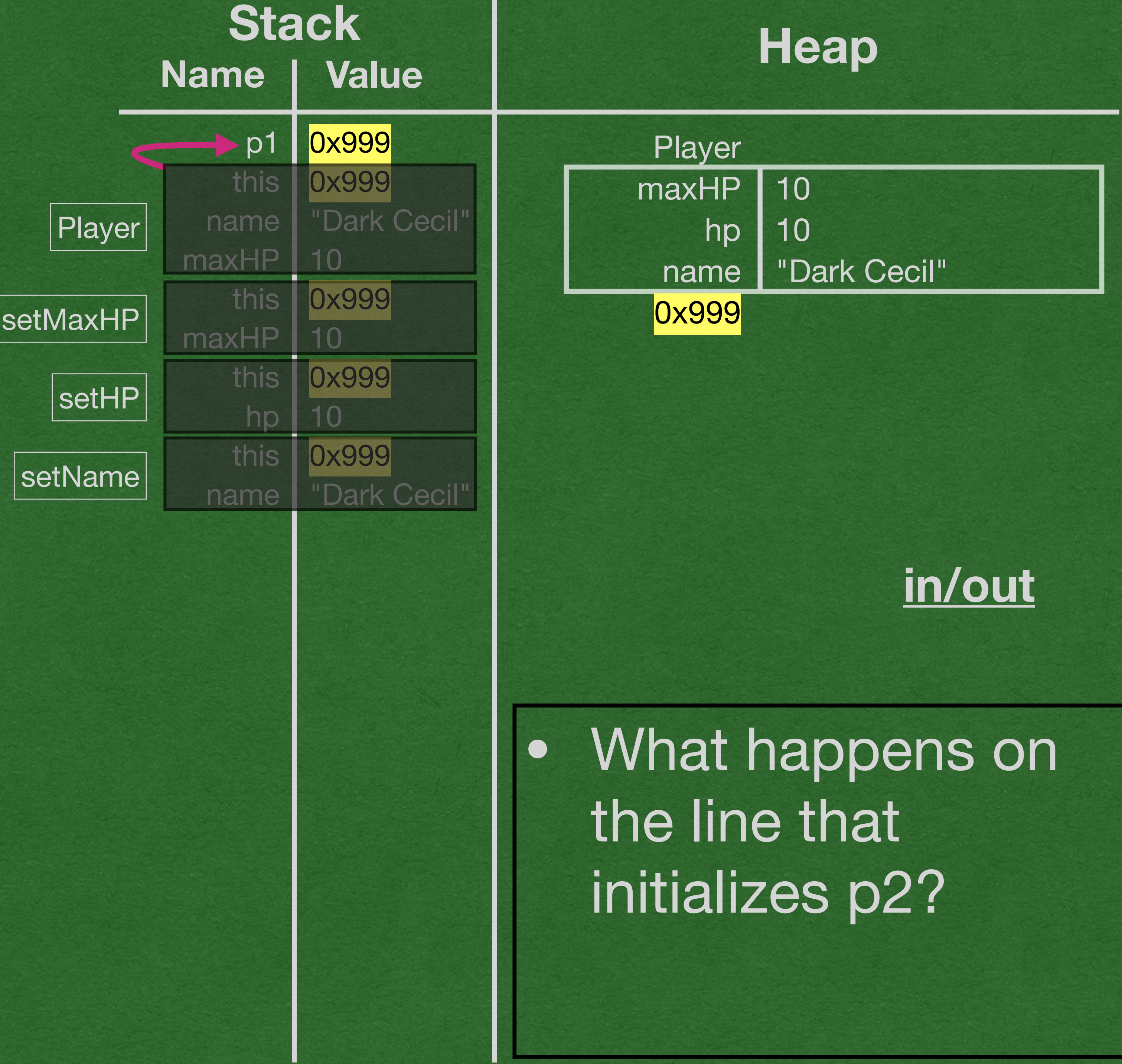
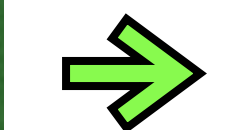
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

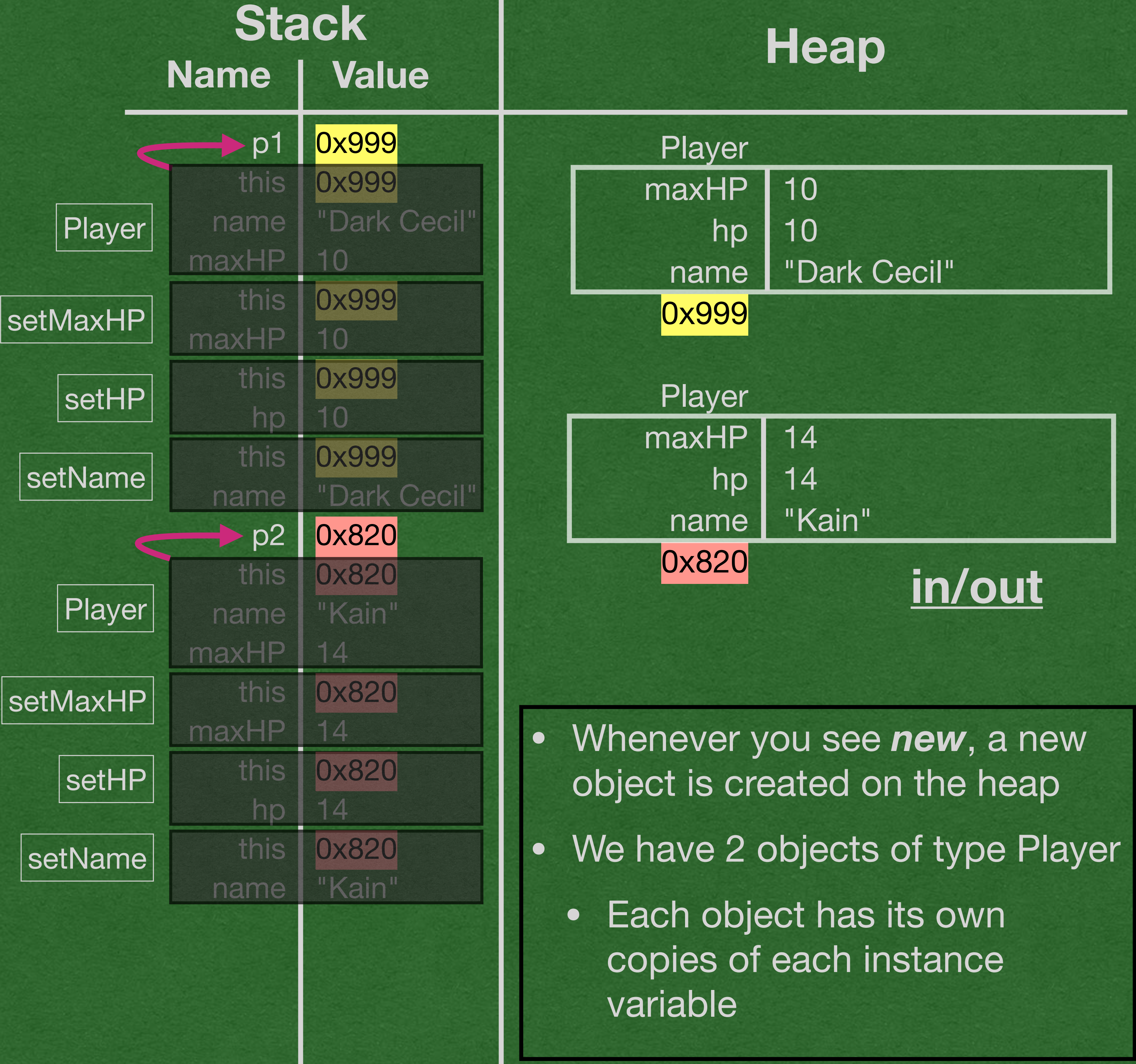
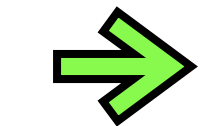
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

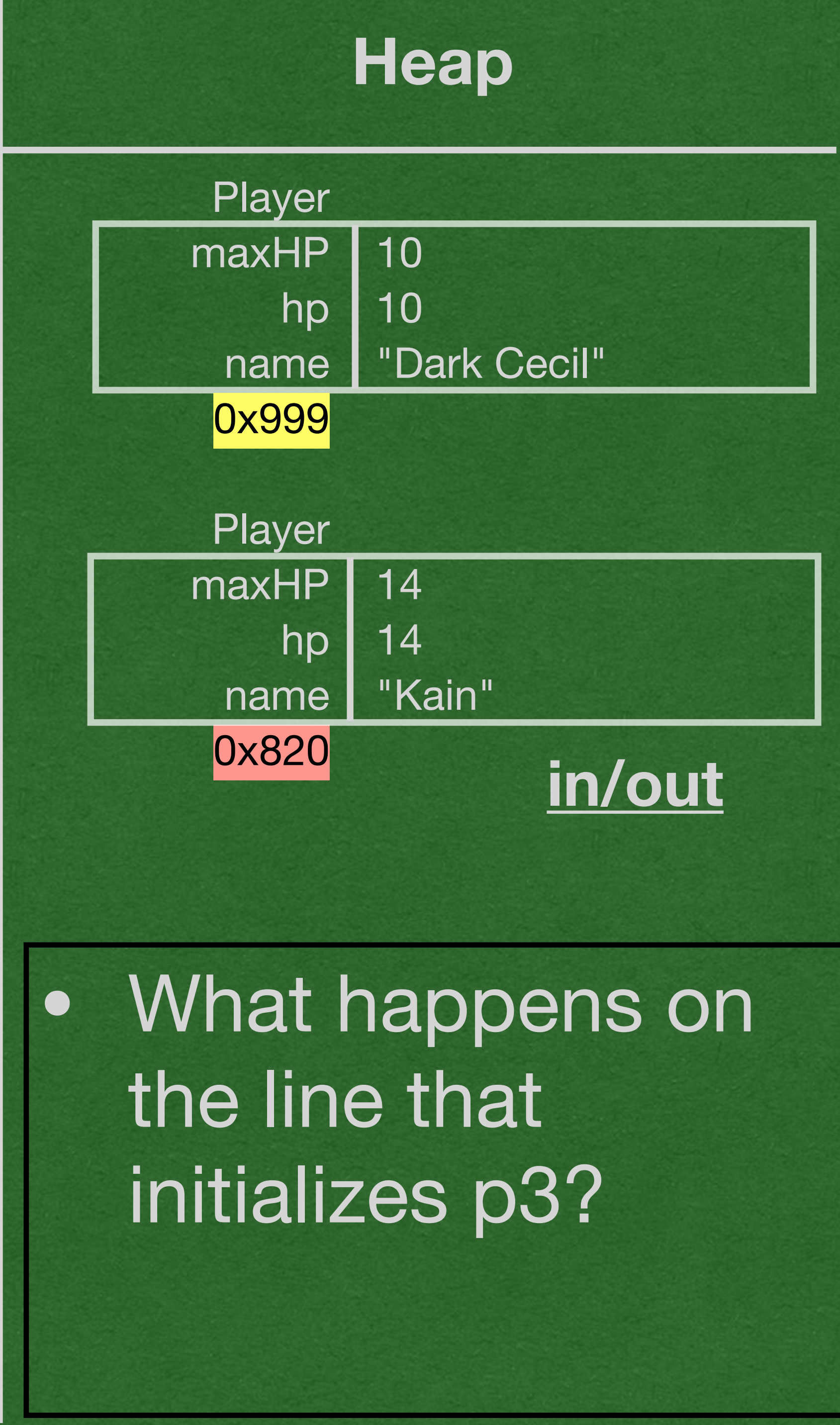
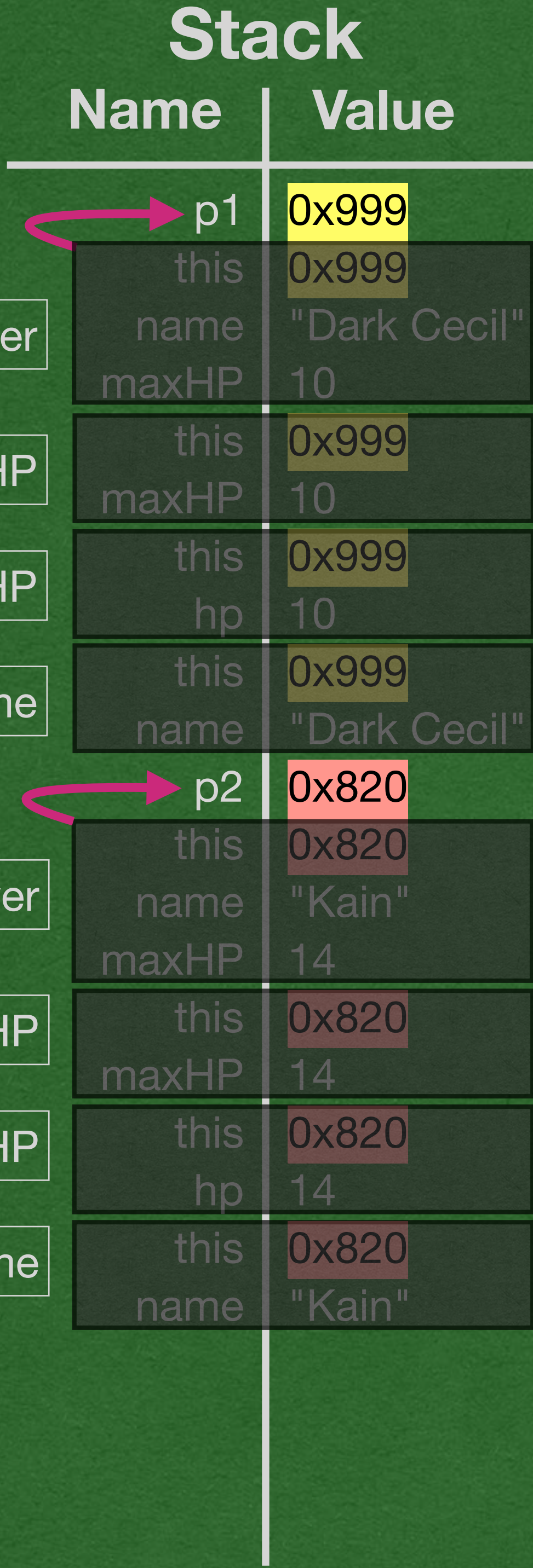
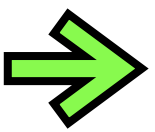
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

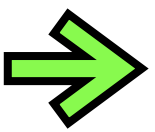
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

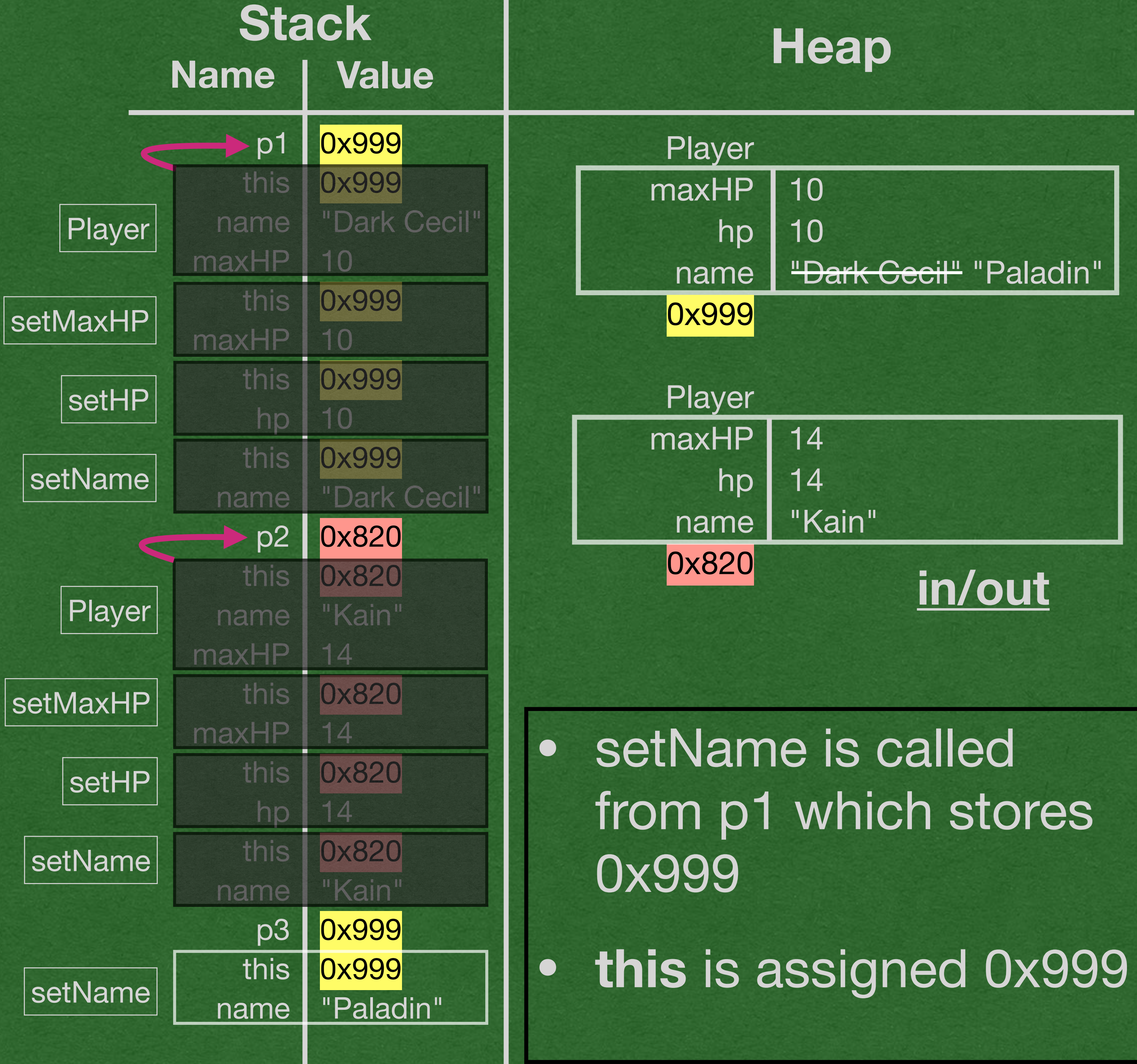
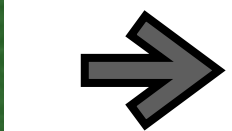
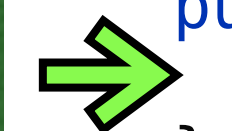
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

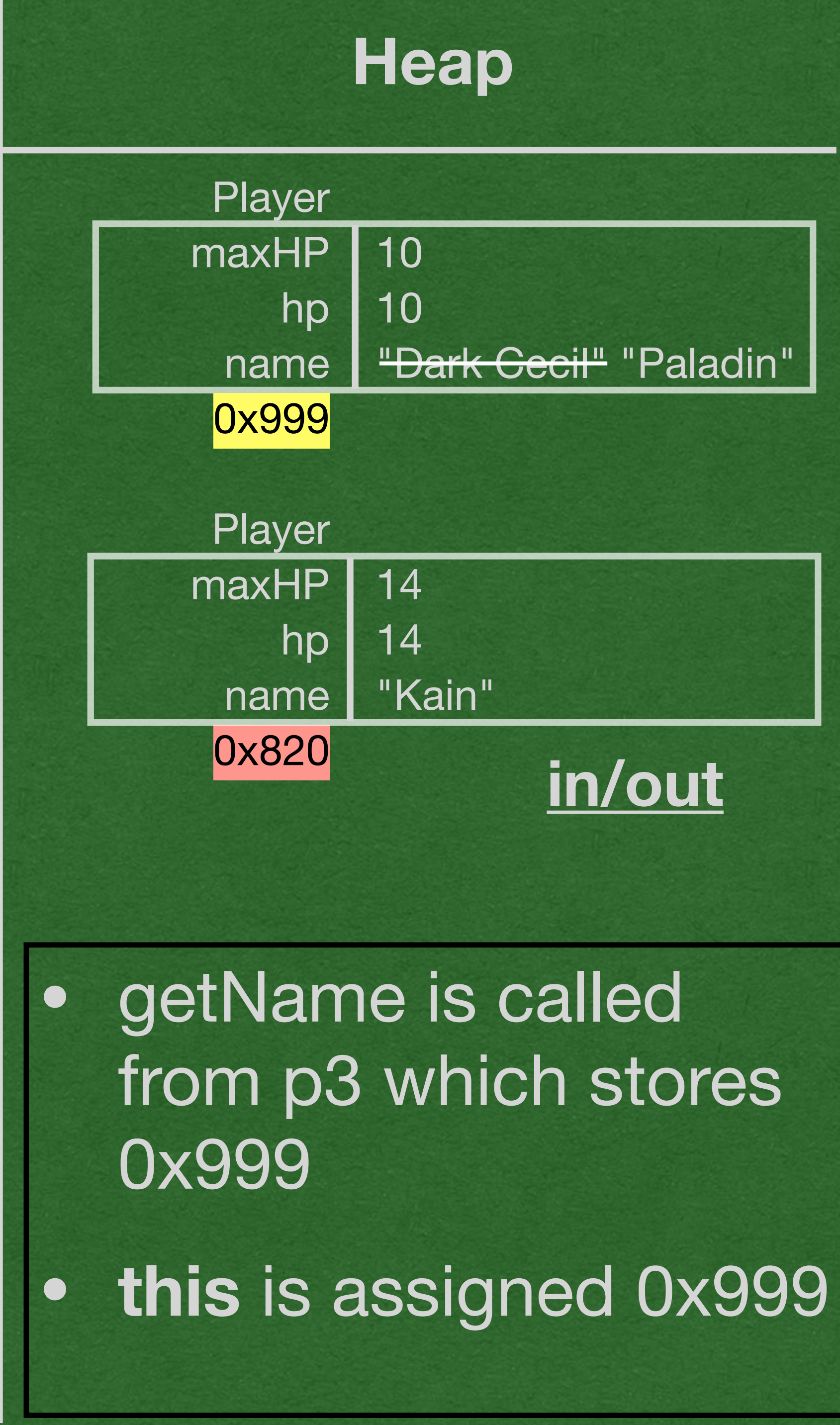
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

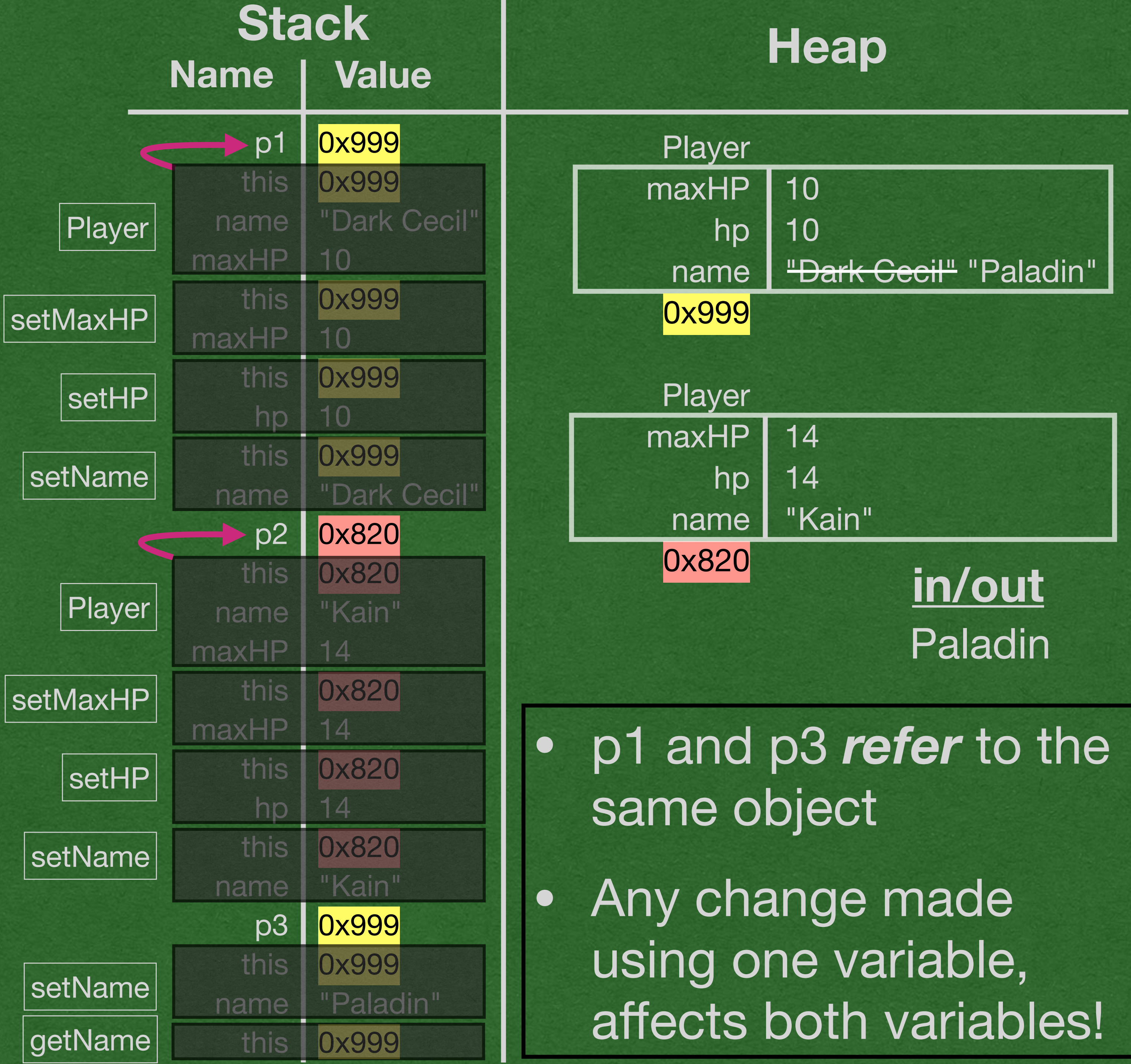
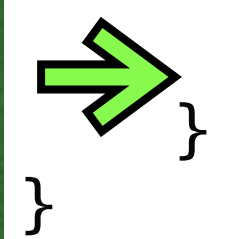
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





```
public class Player {
    private int maxHP;
    private int hp;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

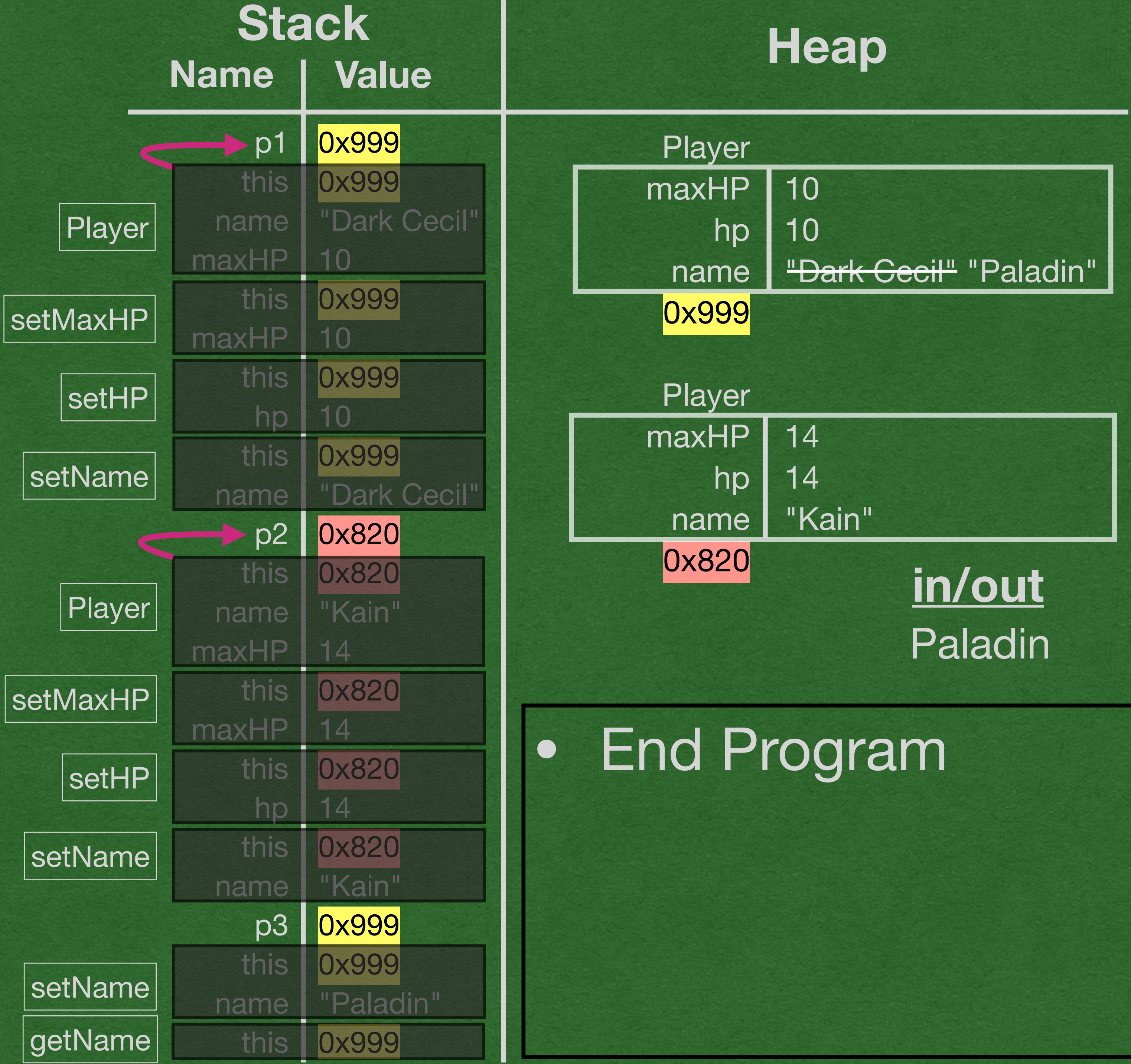
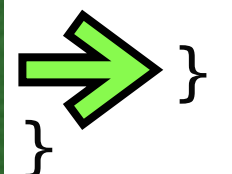
    public void setMaxHP(int maxHP) {
        this.maxHP = maxHP;
    }

    public void setHP(int hp) {
        if (hp <= this.maxHP) {
            this.hp = hp;
        } else {
            this.hp = this.maxHP;
        }
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p1;
        p1.setName("Paladin");
        System.out.println(p3.getName());
    }
}
```





# Stack Memory

- Only primitive types are stored directly on the stack
  - double
  - int
  - char
  - boolean
  - String\*
  - Double/Integer/Character/Boolean\*
- **Everything else** is stored on the heap with only their references stored on the stack\*\*
  - This includes **every** object created from a class that **you wrote**

\*Strings and boxed types are actually more complex, but we will treat them as though they are on the stack in this course because they *behave* exactly as a value on the stack

\*\*Stack and heap allocations vary by compiler and JVM implementations. With modern optimizations, we can never be sure where our values will be stored  
We'll use this simplified view so we can move on and learn Computer Science



# toString

```
package week3;

public class Player {

    private int maxHP;
    private int hp;
    private int attackPower = 4;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

    /** Getters and Setters removed for slide **/

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        System.out.println(p1)
    }
}
```

**week3.Player@279f2327**

- When you print one of your objects to the screen
- It prints garbage
- Fully qualified name
- @
- "random" hex value
- Almost always not what you want



# toString

```
package week3;

public class Player {

    private int maxHP;
    private int hp;
    private int attackPower = 4;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

    public String toString() {
        String out = "health: " + this.hp + "/";
        out += this.maxHP;
        return out;
    }

    /** Getters and Setters removed for slide **/

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        System.out.println(p1);
    }
}
```

**health: 10/10**

- If we write a special method named "toString" that returns a String
- This method will be called when we print an object of this type



# Types

- Classes define types
- Now that we have a Player type, we can use it wherever we need a type
- Here, we use Player as the type of a method parameter

```
public class Player {
    private int maxHP;
    private int hp;
    private int attackPower = 4;
    private String name;

    public Player(String name, int maxHP) {
        this.setMaxHP(maxHP);
        this.setHP(maxHP);
        this.setName(name);
    }

    public String toString() {
        String out = "health: " + this.hp + "/";
        out += this.maxHP;
        return out;
    }

    public void takeDamage(int damage) {
        this.hp -= damage;
    }

    public void attack(Player otherPlayer) {
        otherPlayer.takeDamage(this.attackPower);
    }

    /** Getters and Setters removed for slide */

    public static void main(String[] args) {
        Player p1 = new Player("Dark Cecil", 10);
        Player p2 = new Player("Kain", 14);
        Player p3 = p2;
        p1.attack(p2);
        p1.attack(p2);
    }
}
```



# Party Example



# Party

```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```

- Let's look at another example
- We'll highlight 2 new concepts with this code:
- Composition
- Garbage Collection



# Composition

- Instance variables of objects can store references to the objects
- The Party class is composed of an ArrayList of Characters

```
public class Character {  
    private int battlesWon = 0;  
    private int expPoints = 0;  
  
    public Character() {}  
    public void winBattle(int xp) {  
        this.battlesWon++;  
        this.expPoints += xp;  
    }  
}
```

```
public class Party {  
    private ArrayList<Character> members;  
    private int battlesWon = 0;  
  
    public Party() {  
        this.members = new ArrayList<>();  
    }  
    public void addCharacter(Character member) {  
        this.members.add(member);  
    }  
    public void winBattle(int xp) {  
        this.battlesWon++;  
        for (int x=0; x < this.members.size(); x++) {  
            this.members.get(x).winBattle(xp);  
        }  
    }  
    public static void main(String[] args) {  
        Character hero = new Character();  
        hero = new Character();  
        Character fighter = new Character();  
        hero.winBattle(10);  
        Party party = new Party();  
        party.addCharacter(hero);  
        party.addCharacter(fighter);  
        party.winBattle(20);  
    }  
}
```



# Garbage Collection

- hero is assigned a reference to an instance/object of type Character
- hero is then reassigned to a new reference
- We no longer have a reference to the first Character object in memory
- Since we cannot access this object, it will be removed from memory

```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



# Memory Diagram

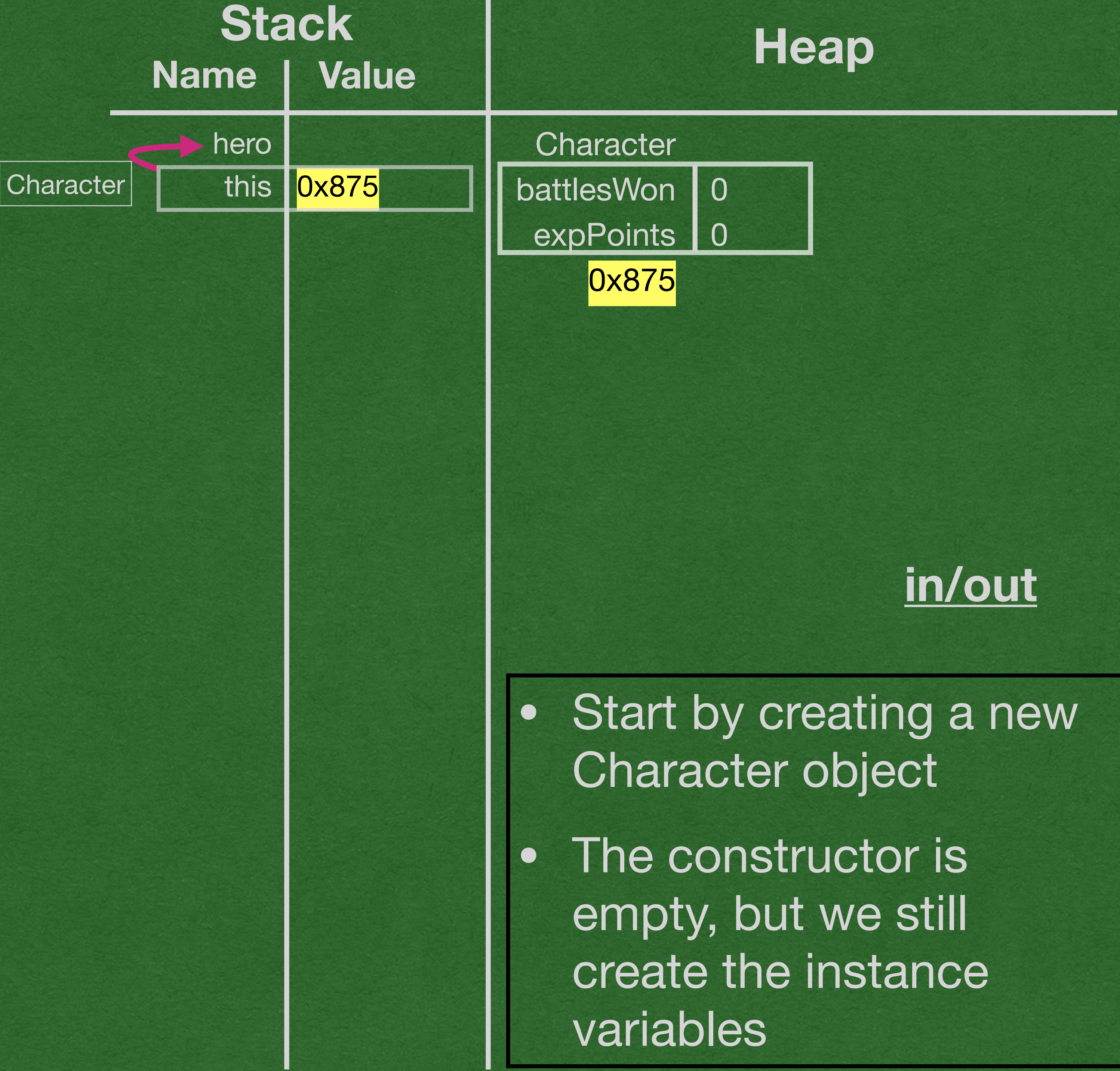


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



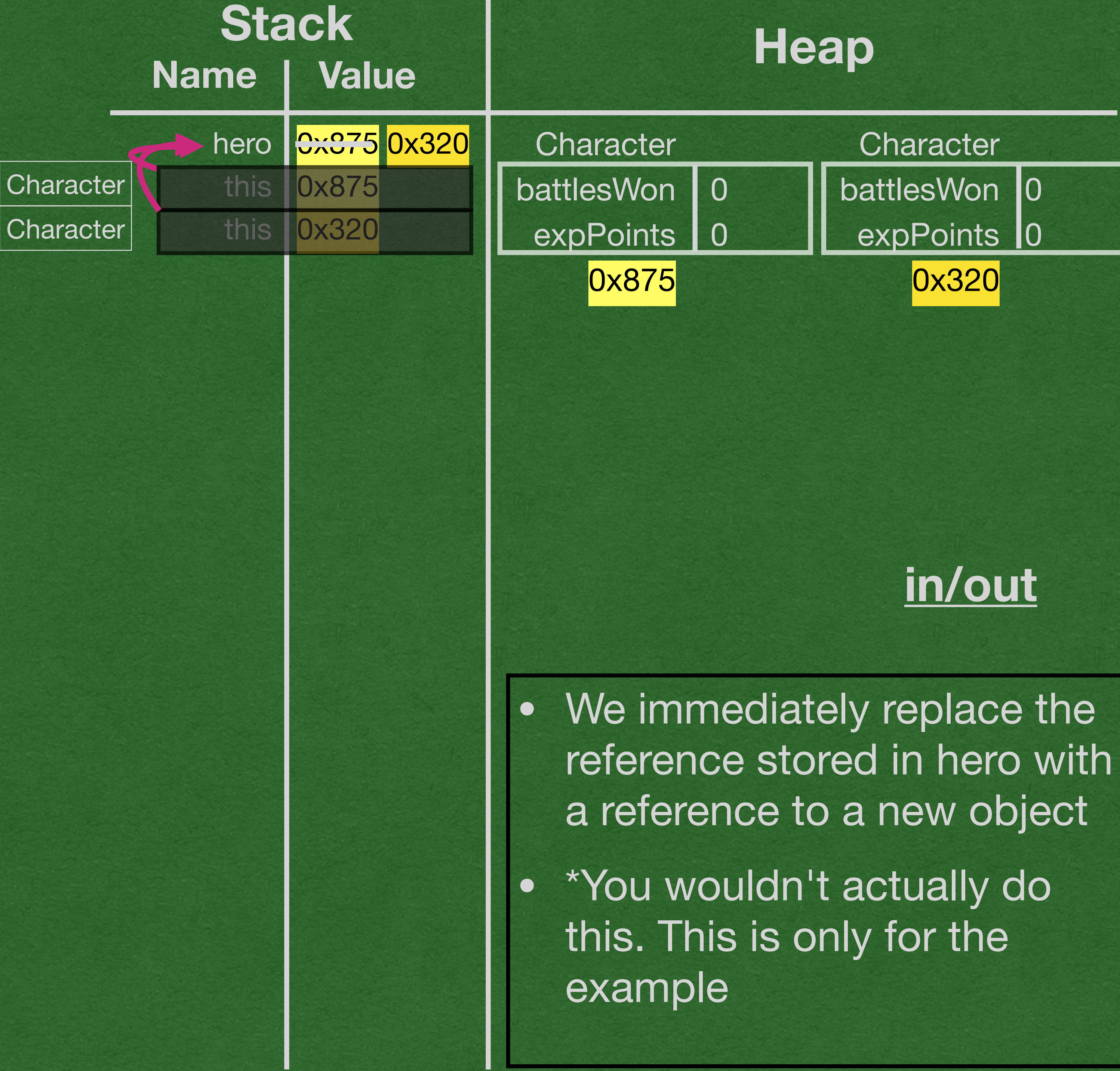
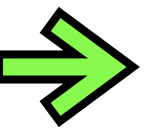


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



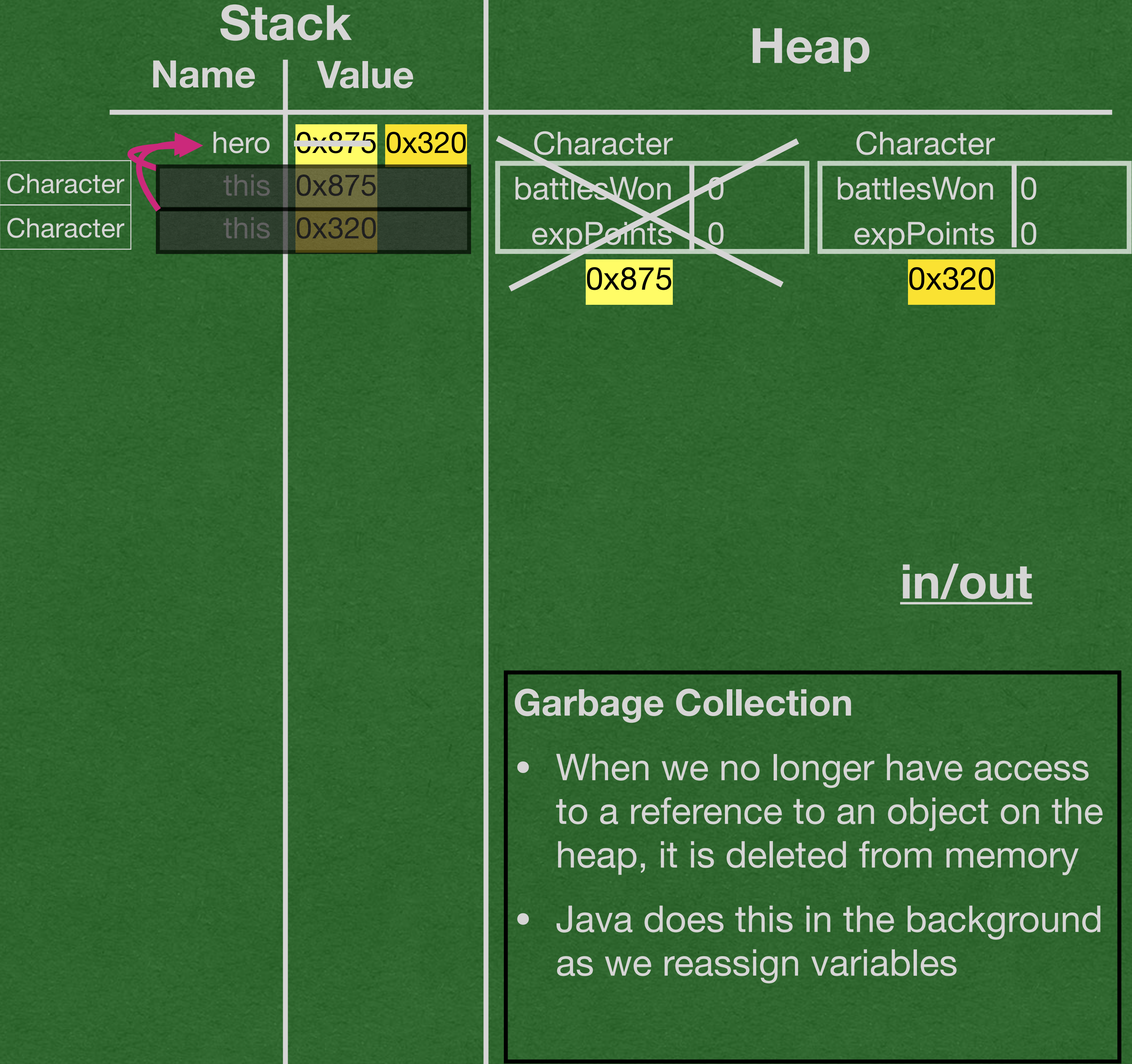
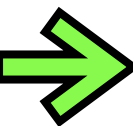


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



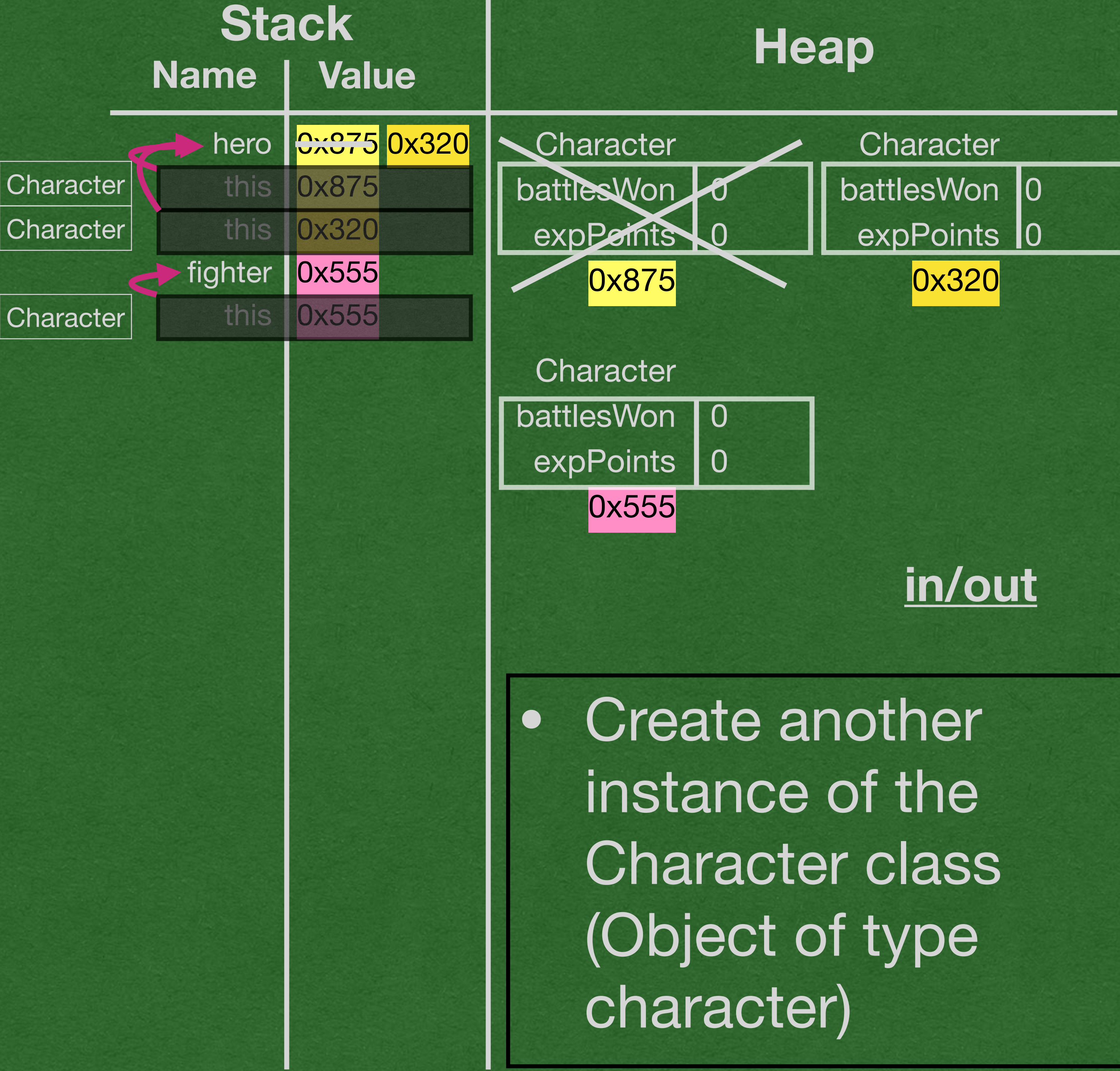
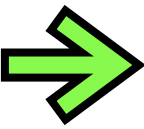


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



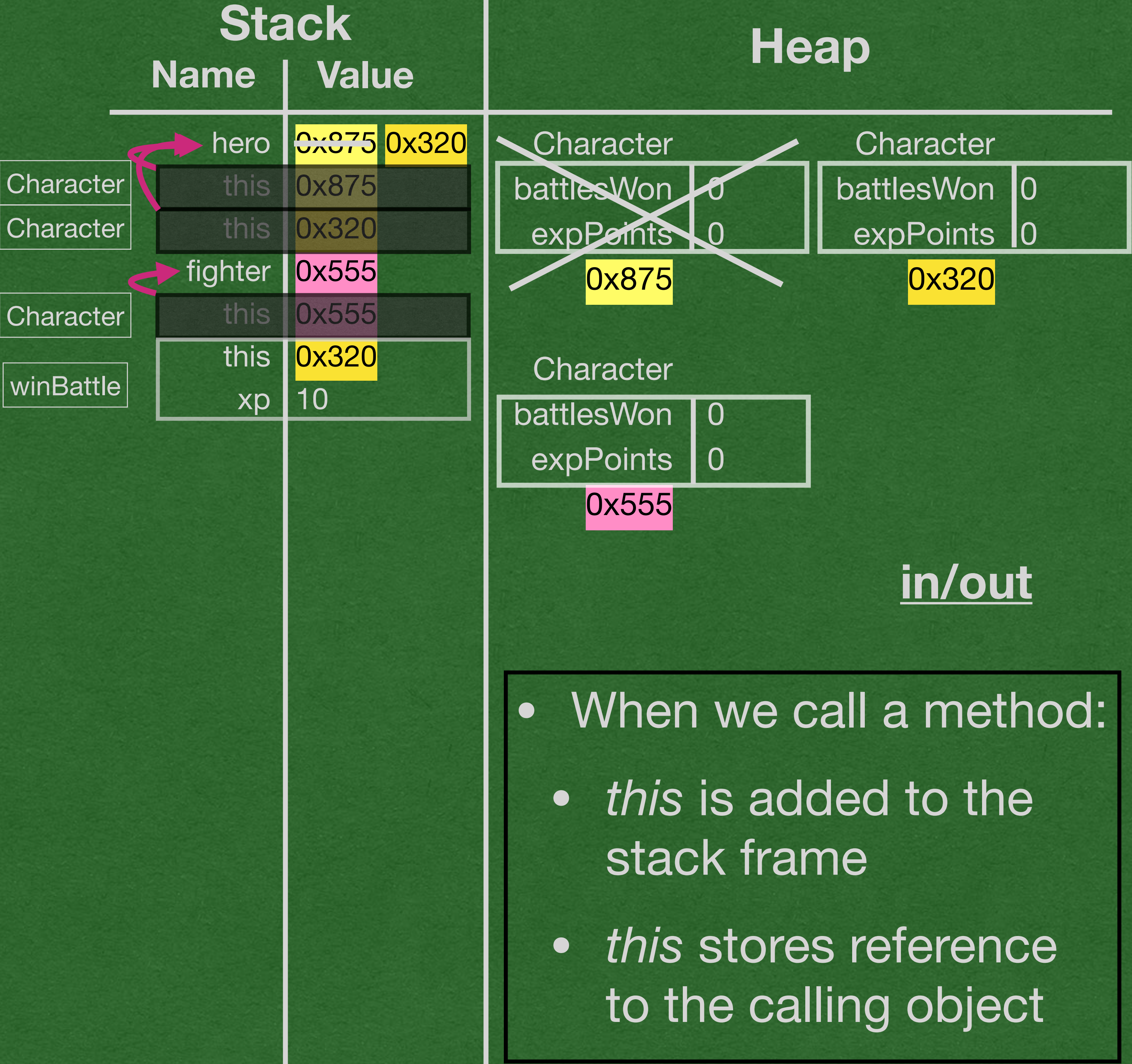


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



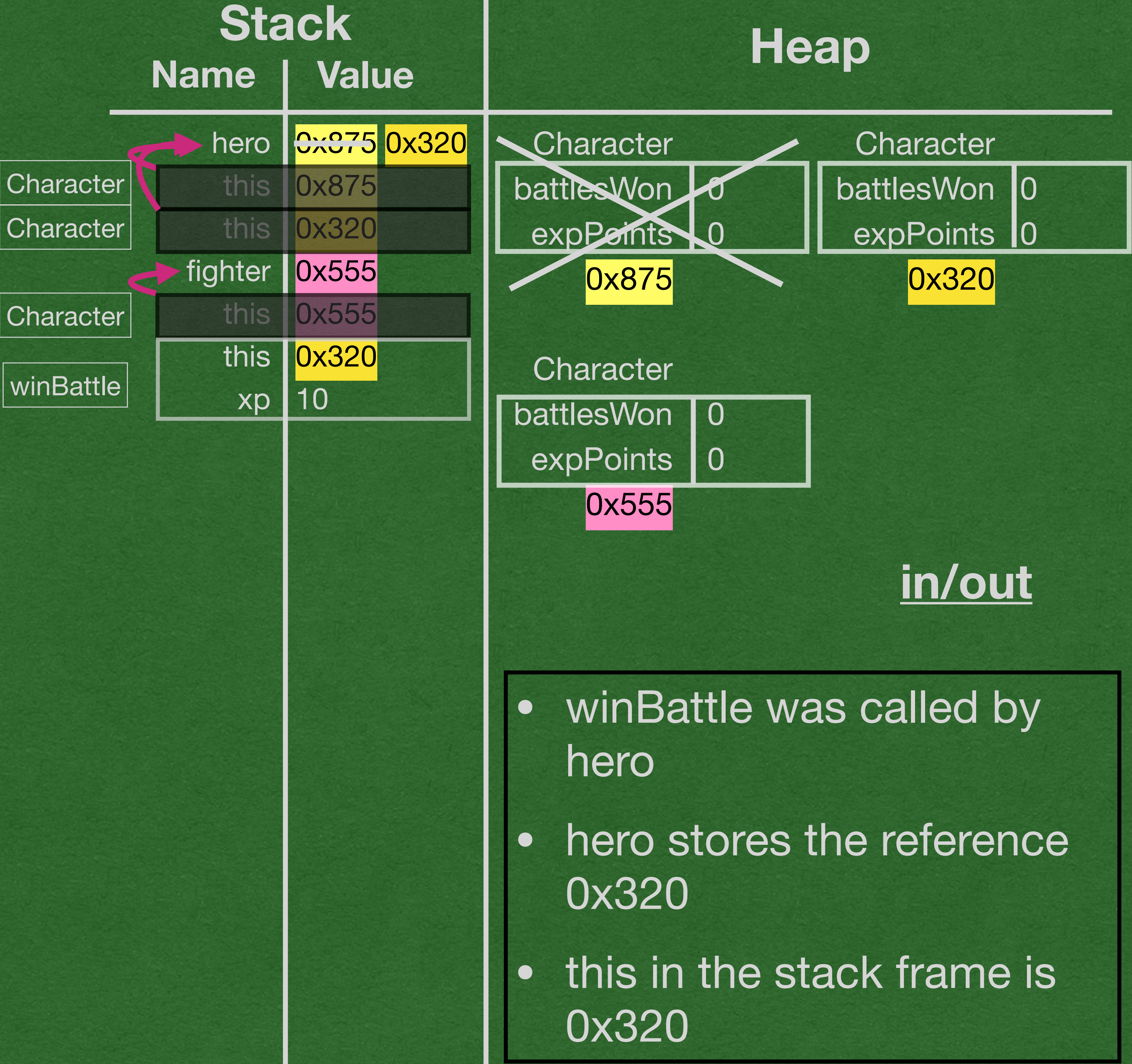


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



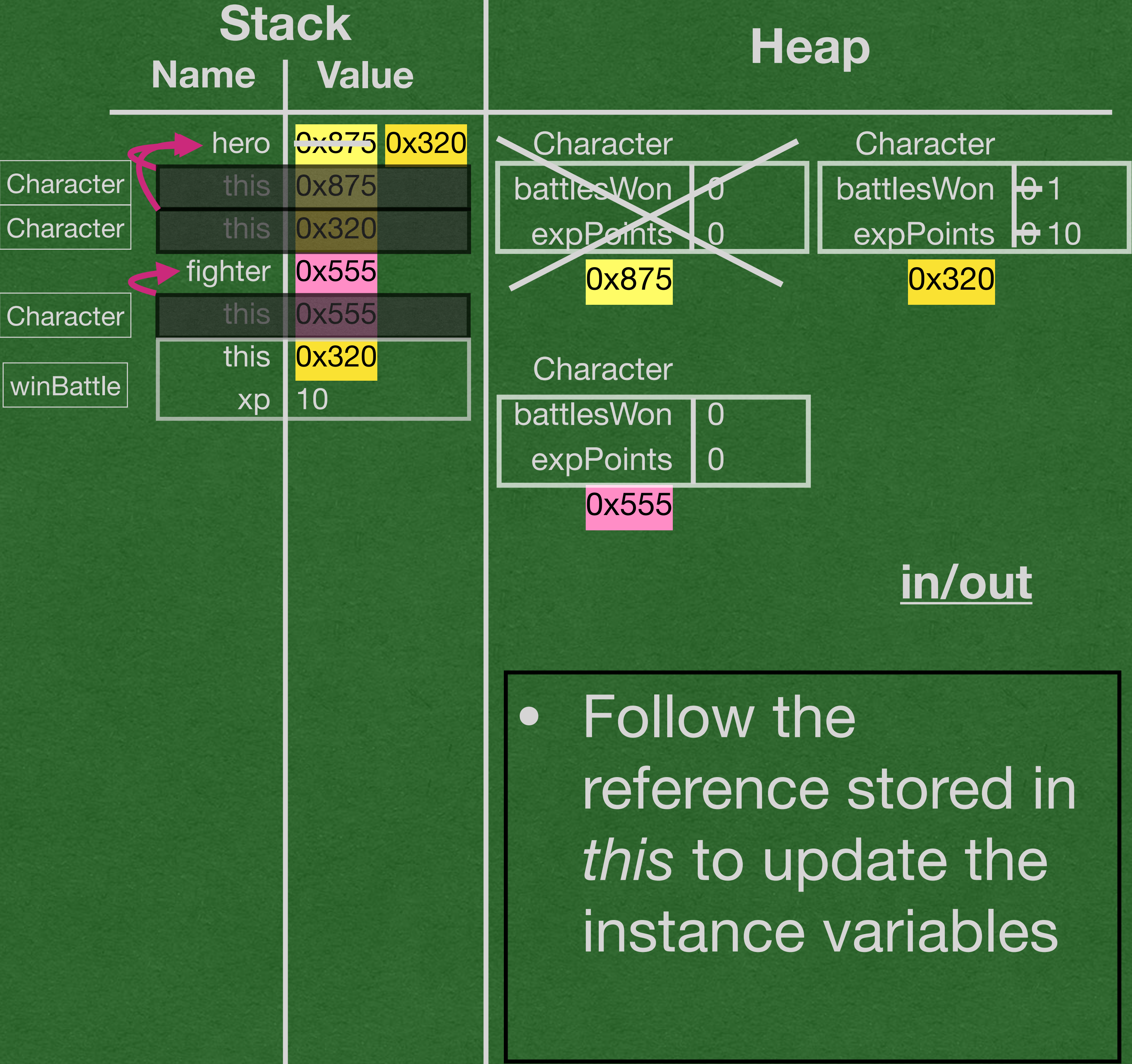


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



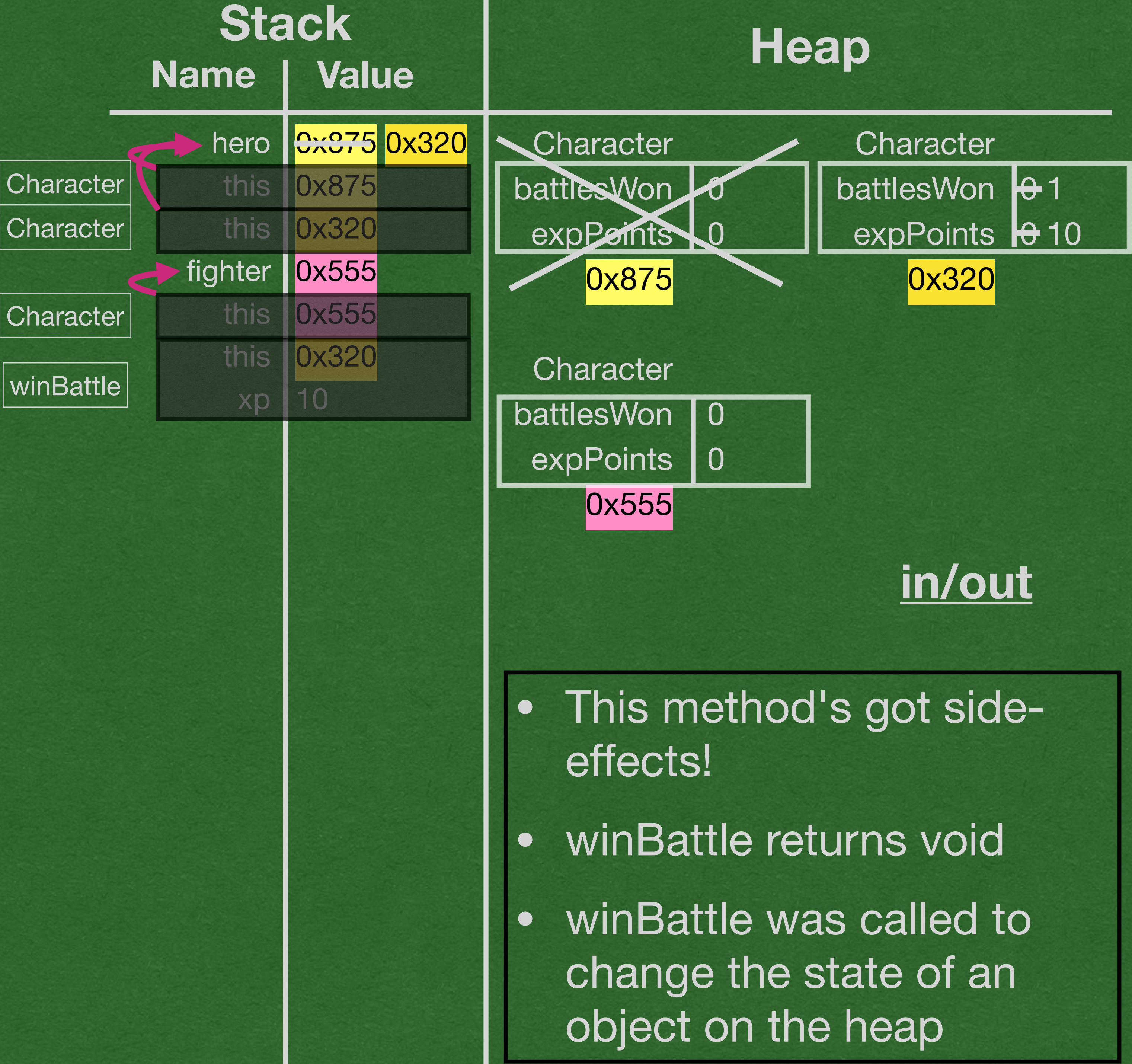


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        ➡ hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



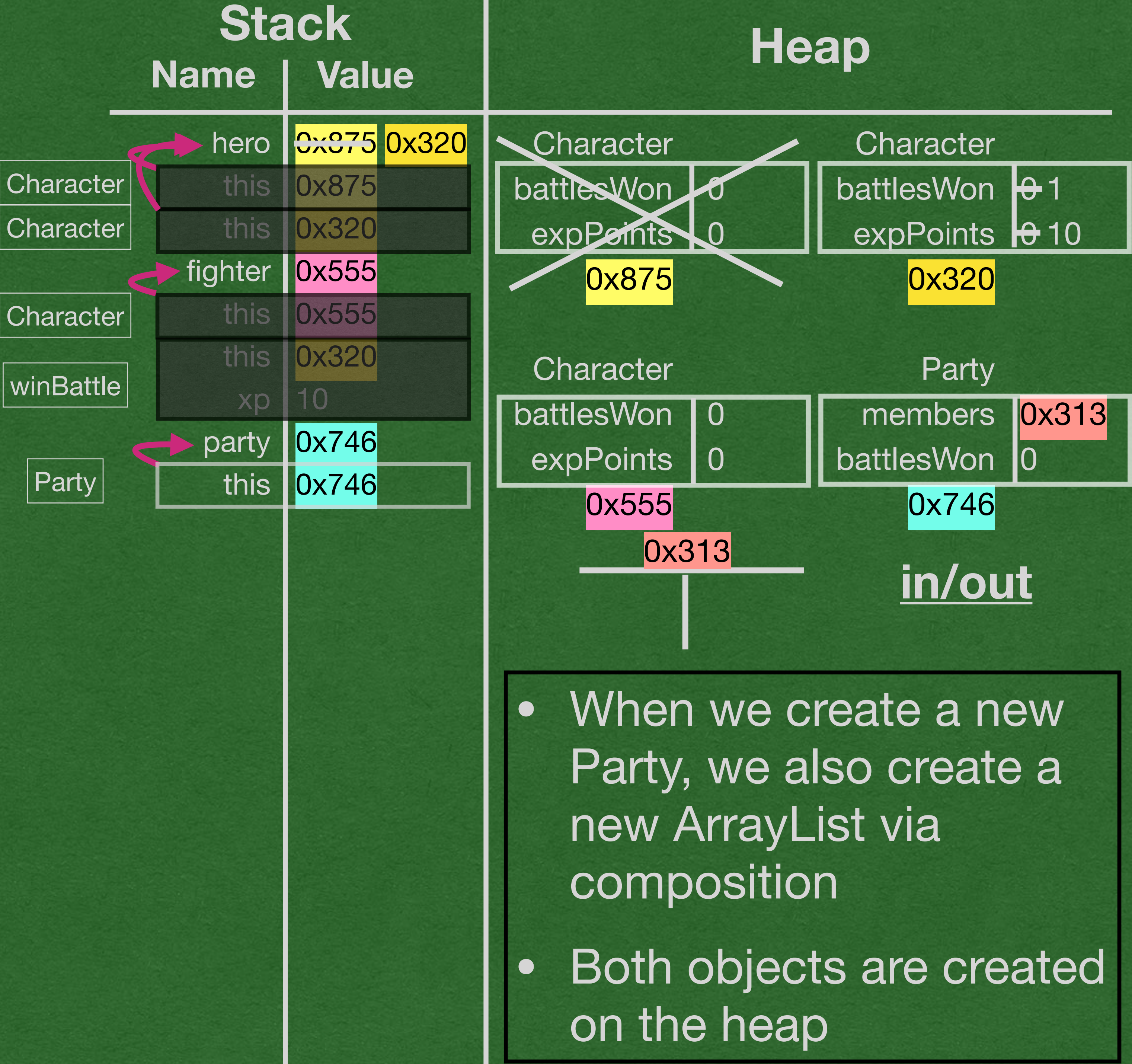


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}

public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



- When we create a new Party, we also create a new ArrayList via composition
- Both objects are created on the heap

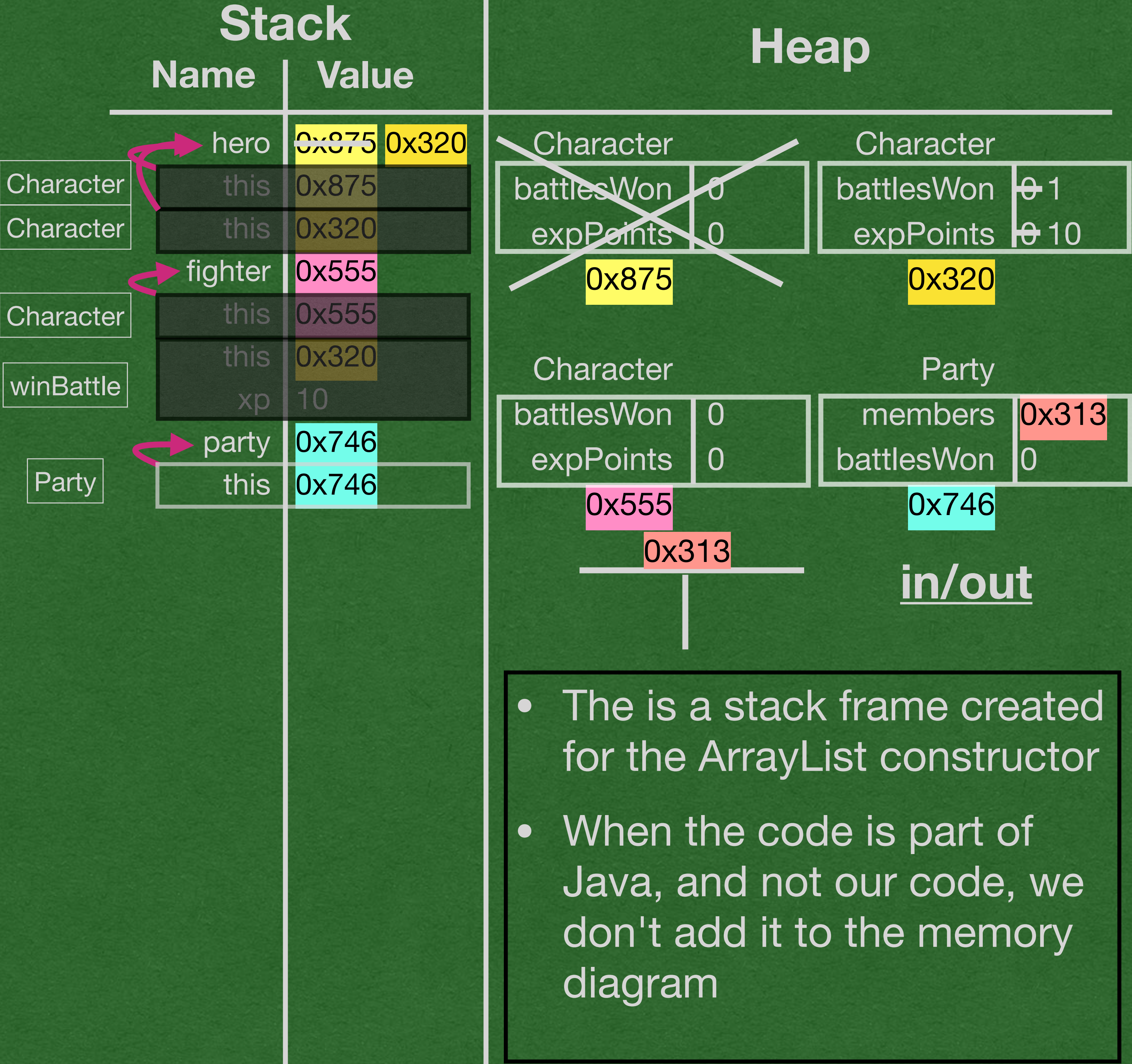


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



- The is a stack frame created for the ArrayList constructor
- When the code is part of Java, and not our code, we don't add it to the memory diagram

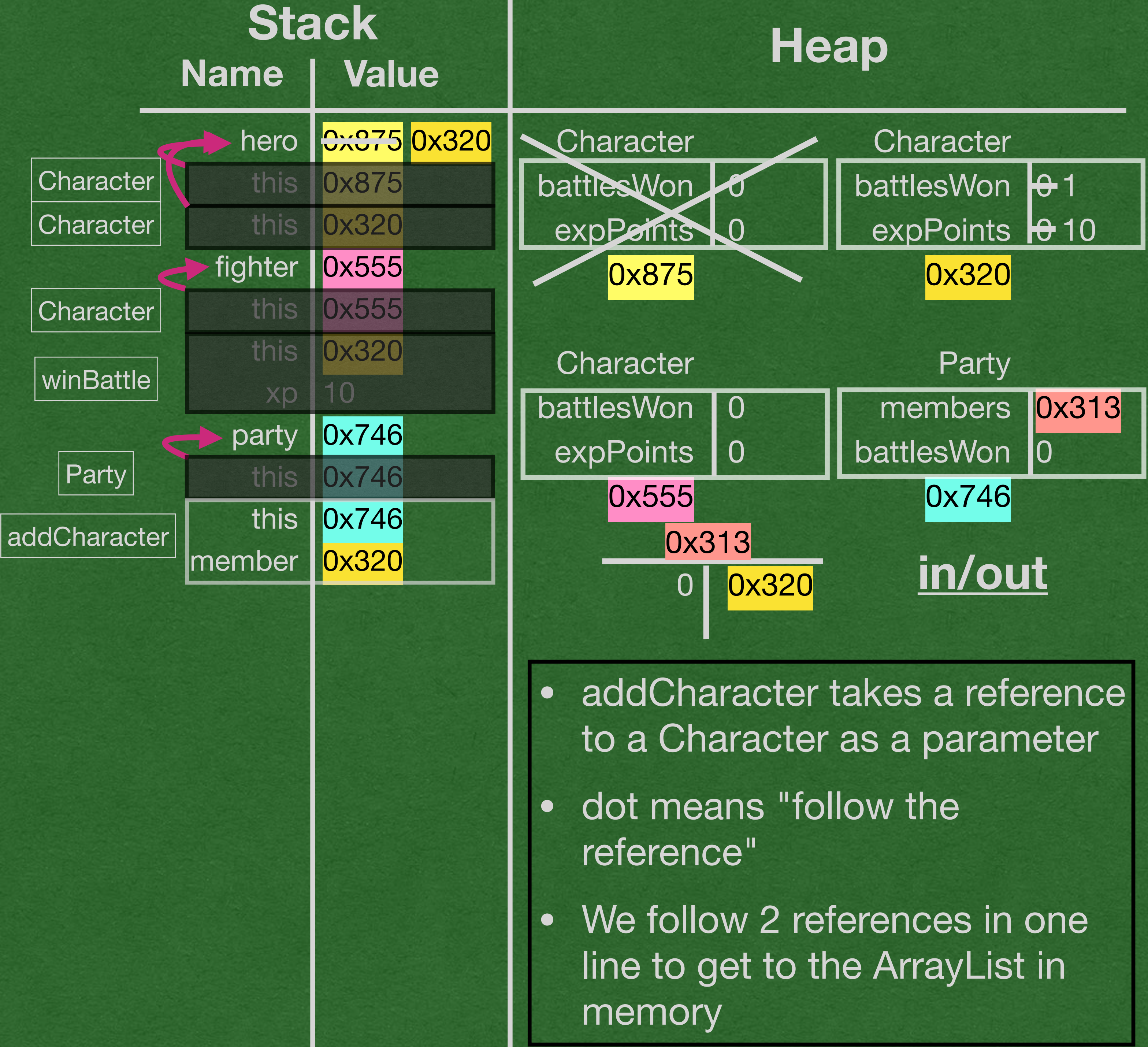


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



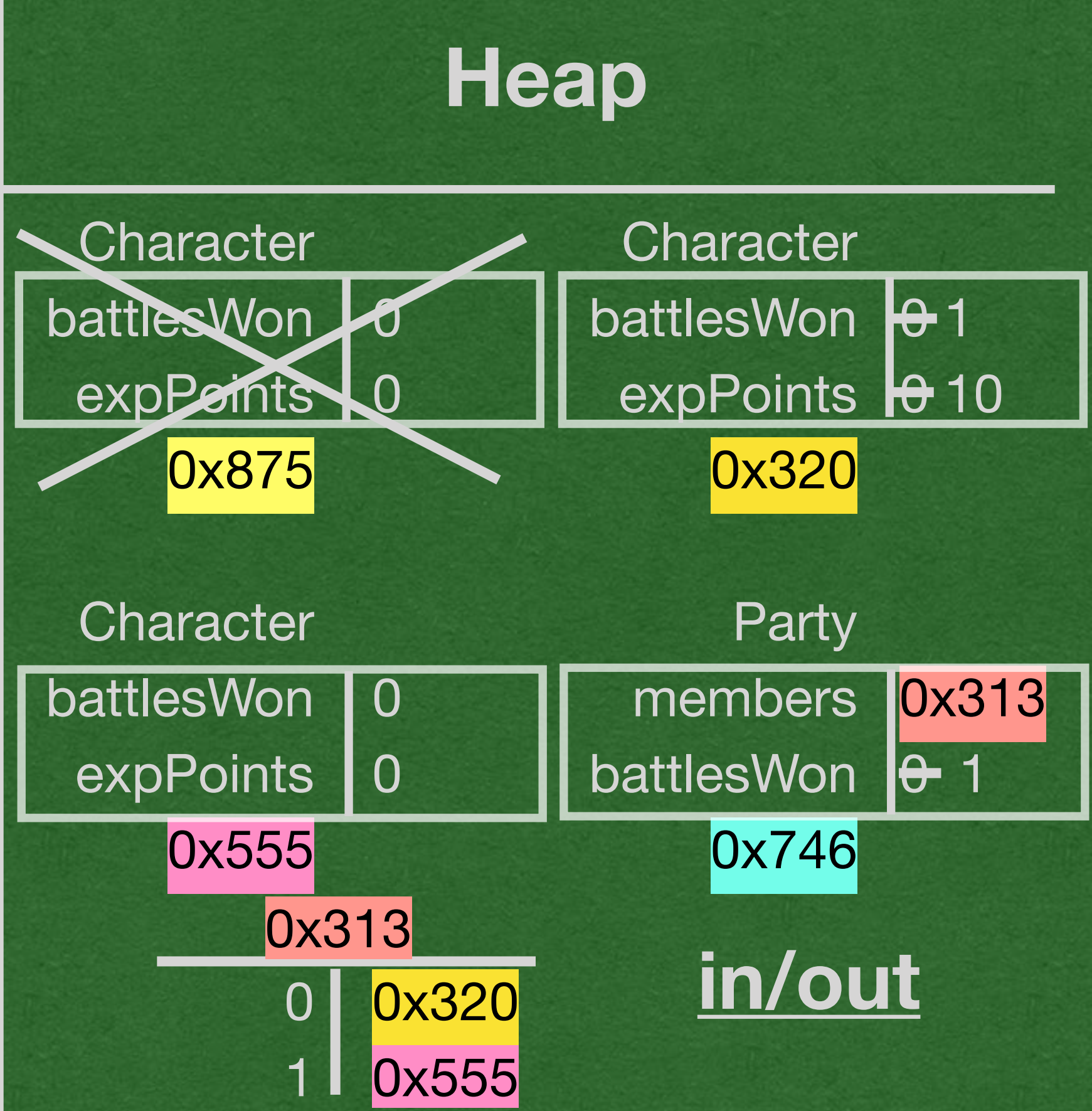
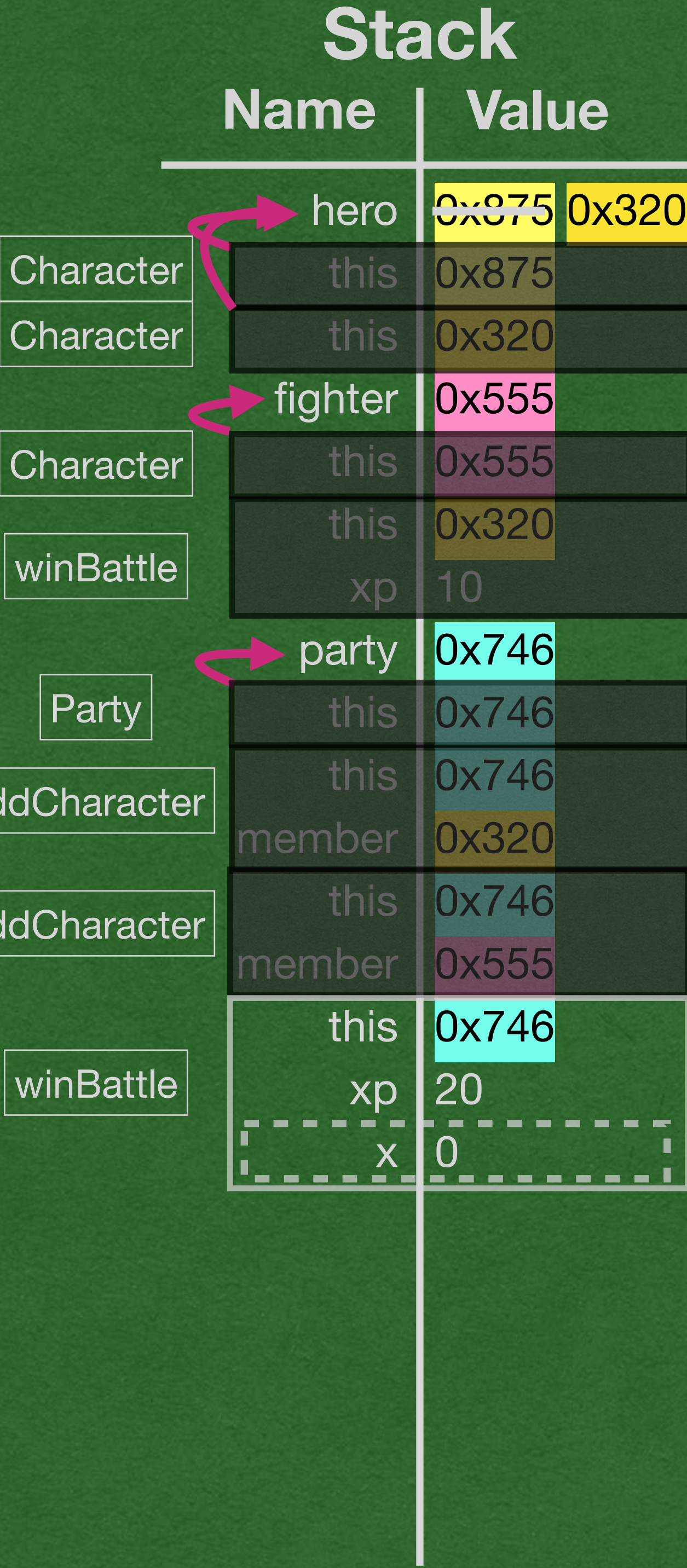
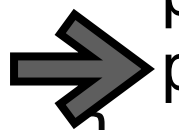


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



- Check out this line!
- dot means follow the reference
- We'll jump around in memory 3 times to reach the Player at 0x320

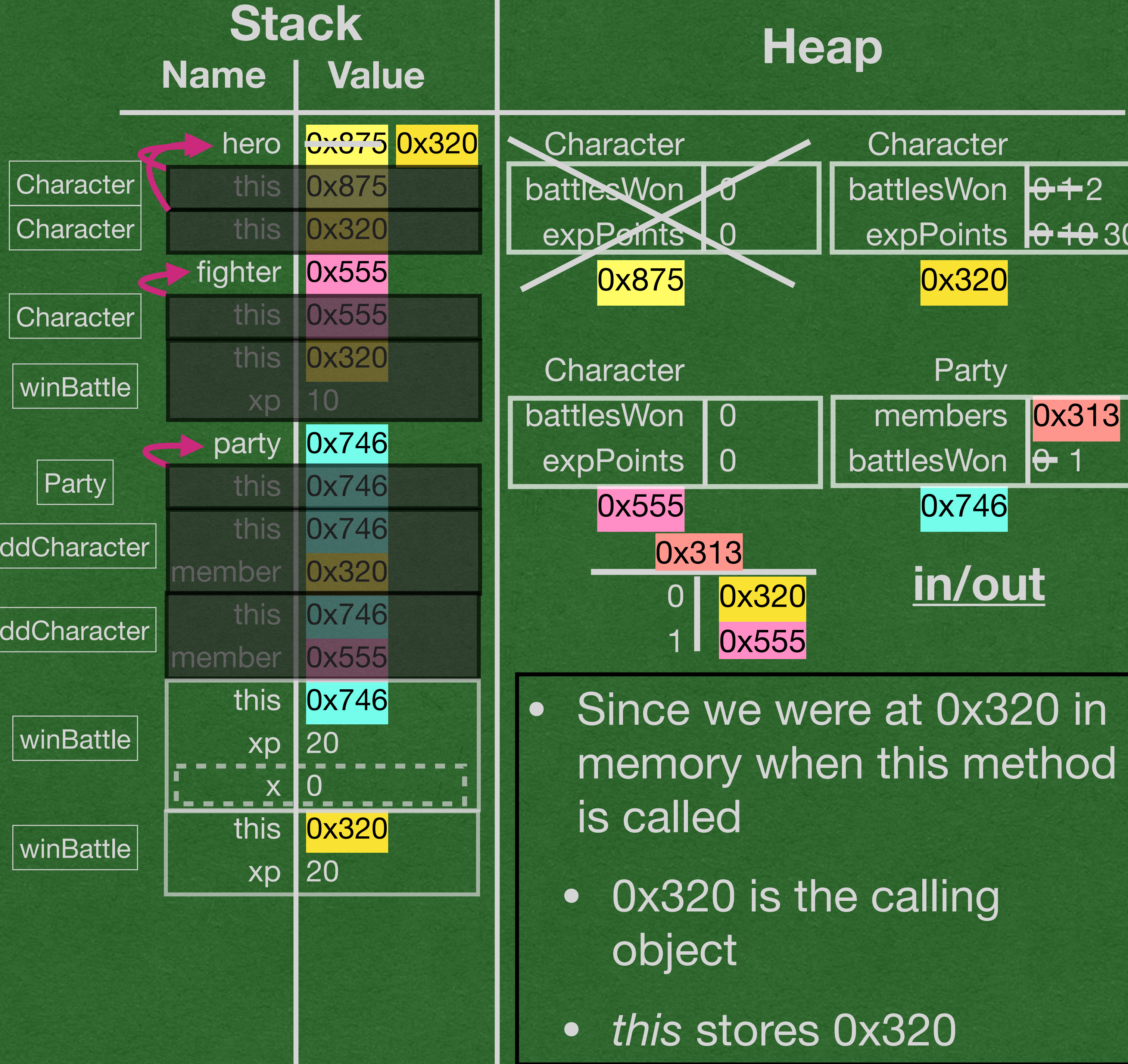


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



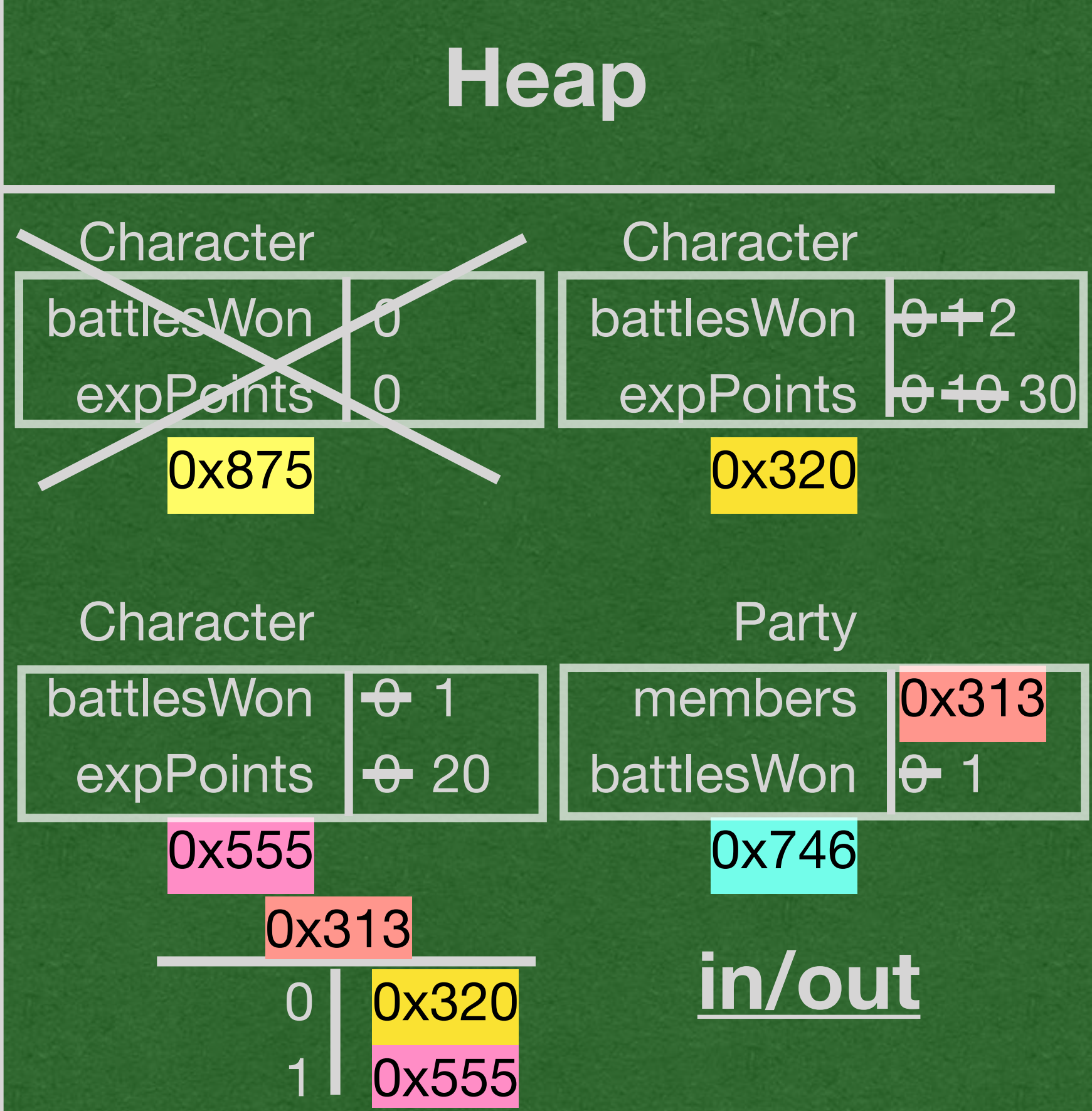
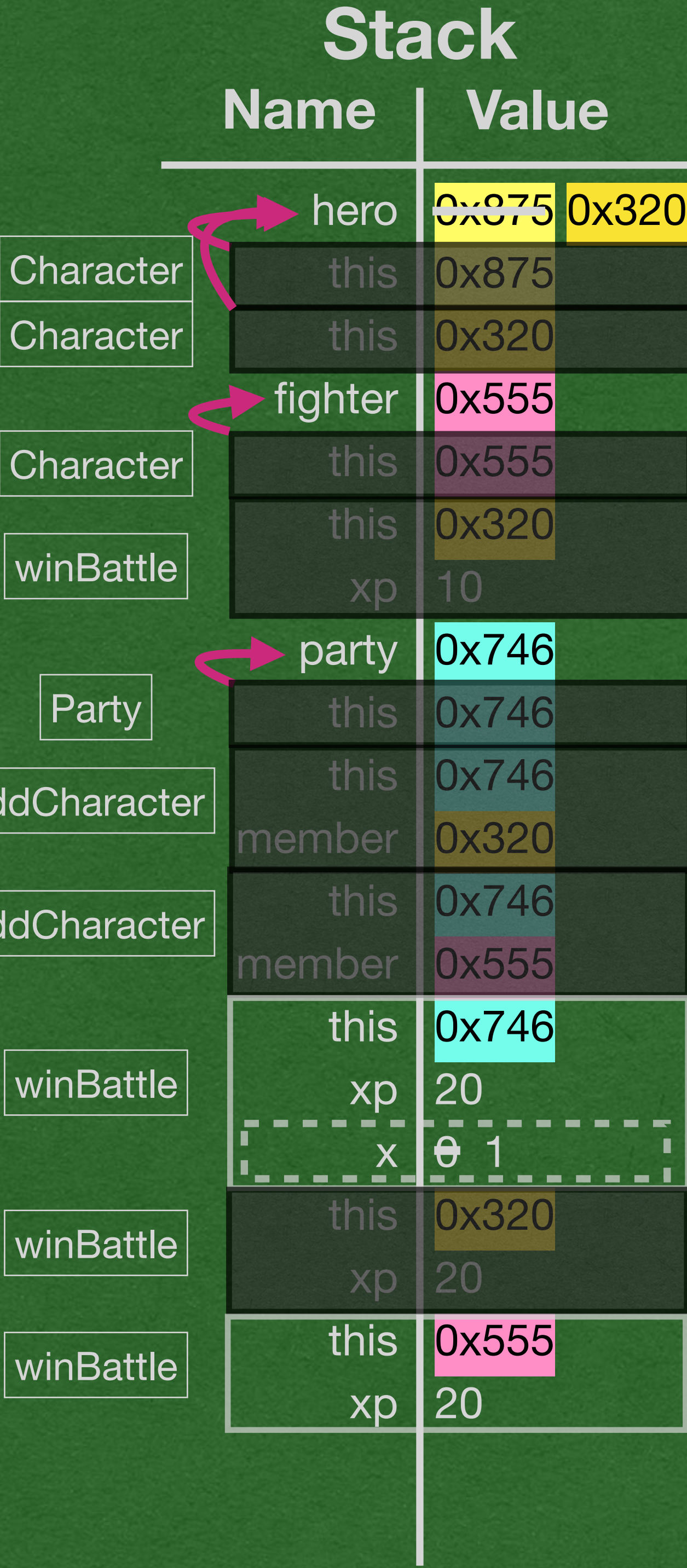


```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



Repeat the process with 0x555



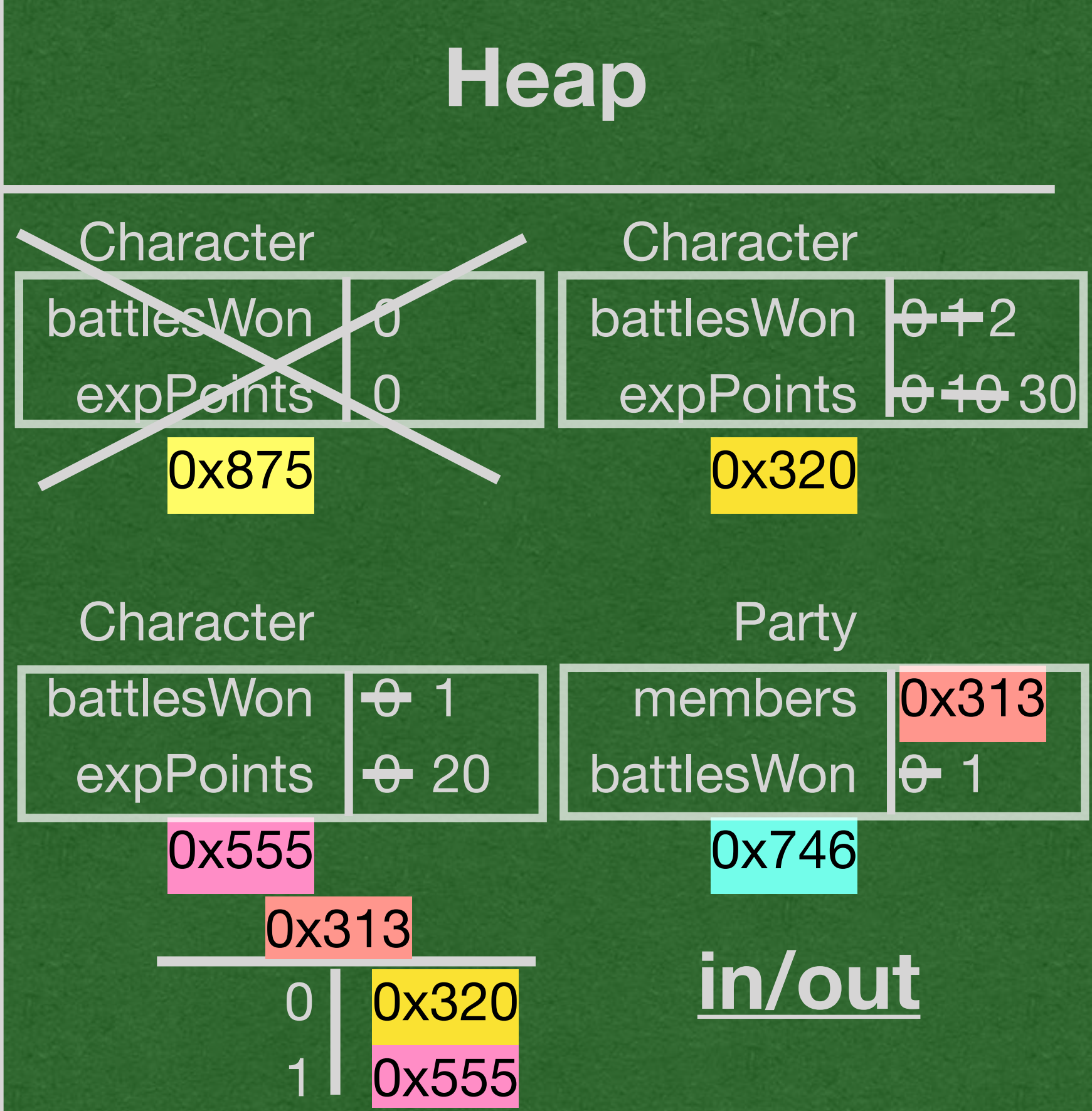
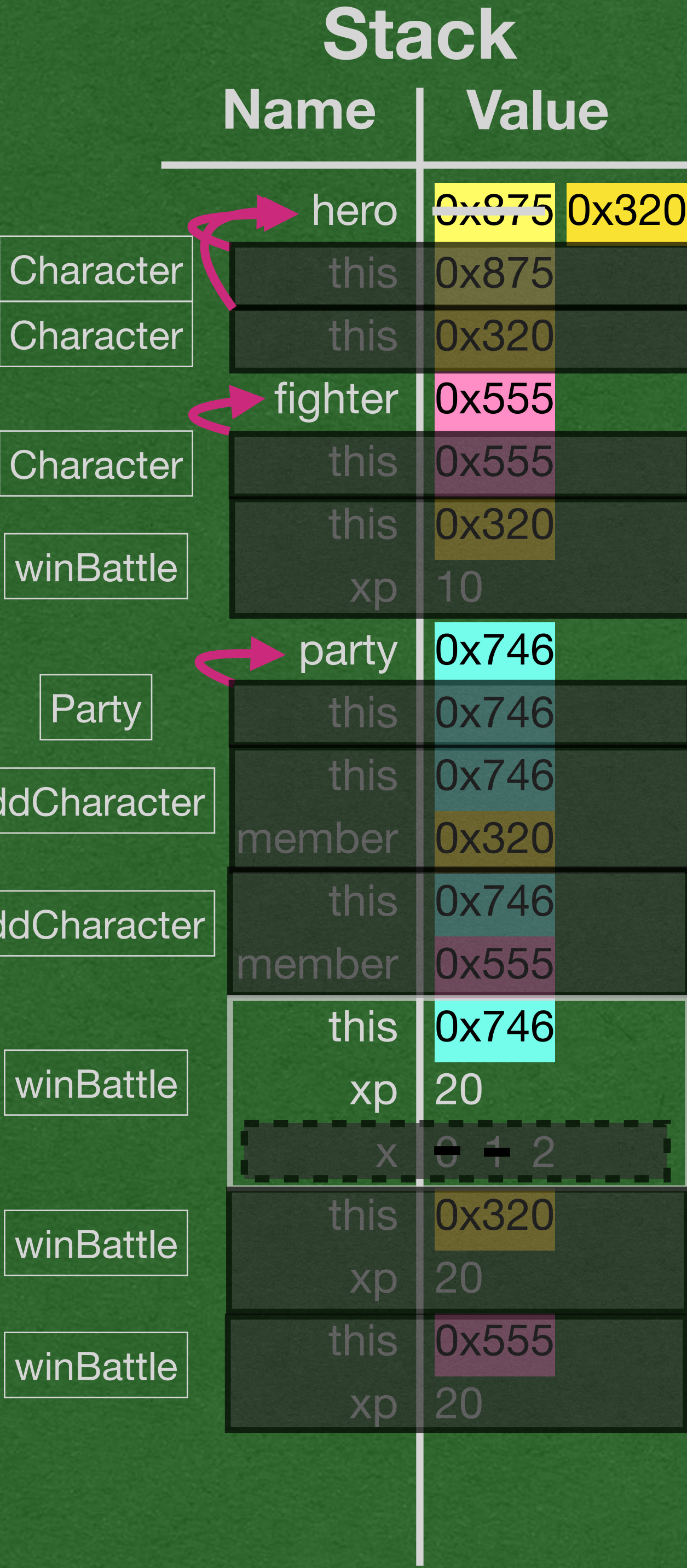
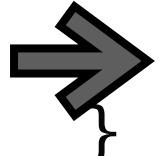
```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }

    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```



- The loop ends when x reaches 2
- x is removed from memory



```
public class Character {
    private int battlesWon = 0;
    private int expPoints = 0;

    public Character() {}
    public void winBattle(int xp) {
        this.battlesWon++;
        this.expPoints += xp;
    }
}
```

```
public class Party {
    private ArrayList<Character> members;
    private int battlesWon = 0;

    public Party() {
        this.members = new ArrayList<>();
    }
    public void addCharacter(Character member) {
        this.members.add(member);
    }
    public void winBattle(int xp) {
        this.battlesWon++;
        for (int x=0; x < this.members.size(); x++) {
            this.members.get(x).winBattle(xp);
        }
    }
    public static void main(String[] args) {
        Character hero = new Character();
        hero = new Character();
        Character fighter = new Character();
        hero.winBattle(10);
        Party party = new Party();
        party.addCharacter(hero);
        party.addCharacter(fighter);
        party.winBattle(20);
    }
}
```

