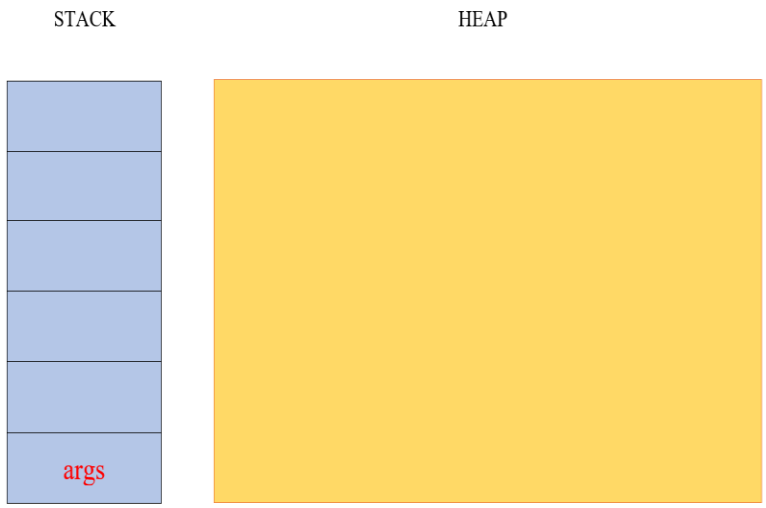
Stack-Heap Diagram

Code

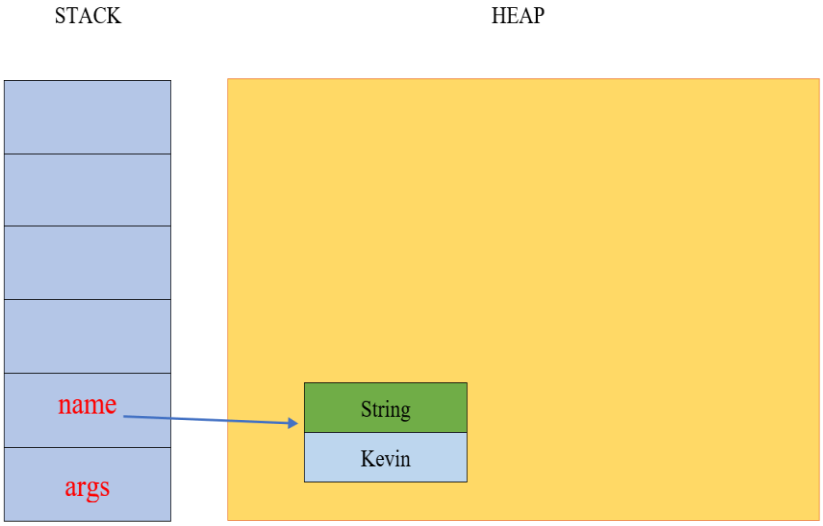
```java
public class Main {
    public static void main(String[] args) {
        String name = "Kevin";
        List<String> list = new ArrayList<>();
        int times = 10;
        System.out.println(times + fill(list, name + name, times));
    }
    public static int fill(Collection<String> collection, String str, int times){
        String shrunk = shrink(str);
        times = (times + shrunk.length()) / 2;
        for (int i = 0; i < times / 2; i++) {
            collection.add(shrunk);
        }
        return times;
    }

    public static String shrink(String str){
        int newLength = str.length() / 2 + str.length() % 2;
        char[] chars = new char[newLength];
        for (int i = 0; i < str.length(); i+=2) {
            chars[i / 2] = str.charAt(i);
        }
        return new String(chars);
    }
}
```

public class Main {

public static void main(String[] args) {

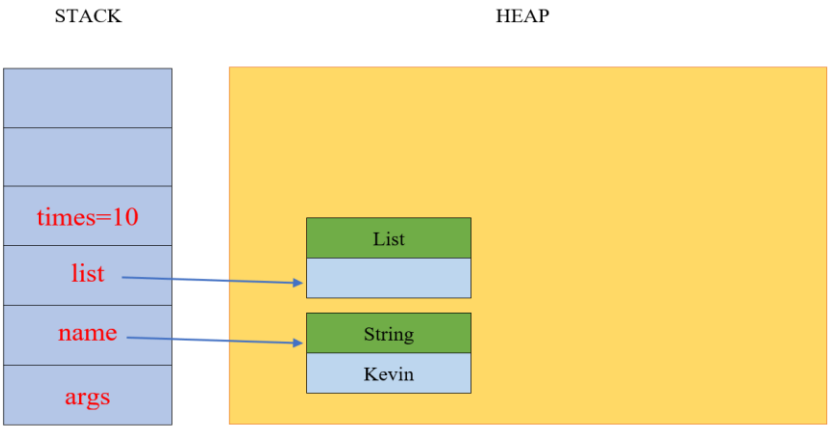


String name = "Kevin";



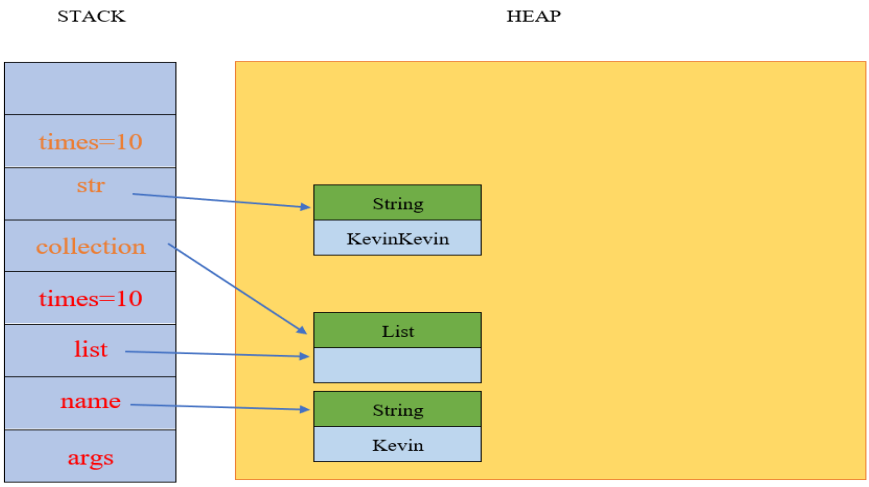
List<String> list = new ArrayList<>();

int times = 10;



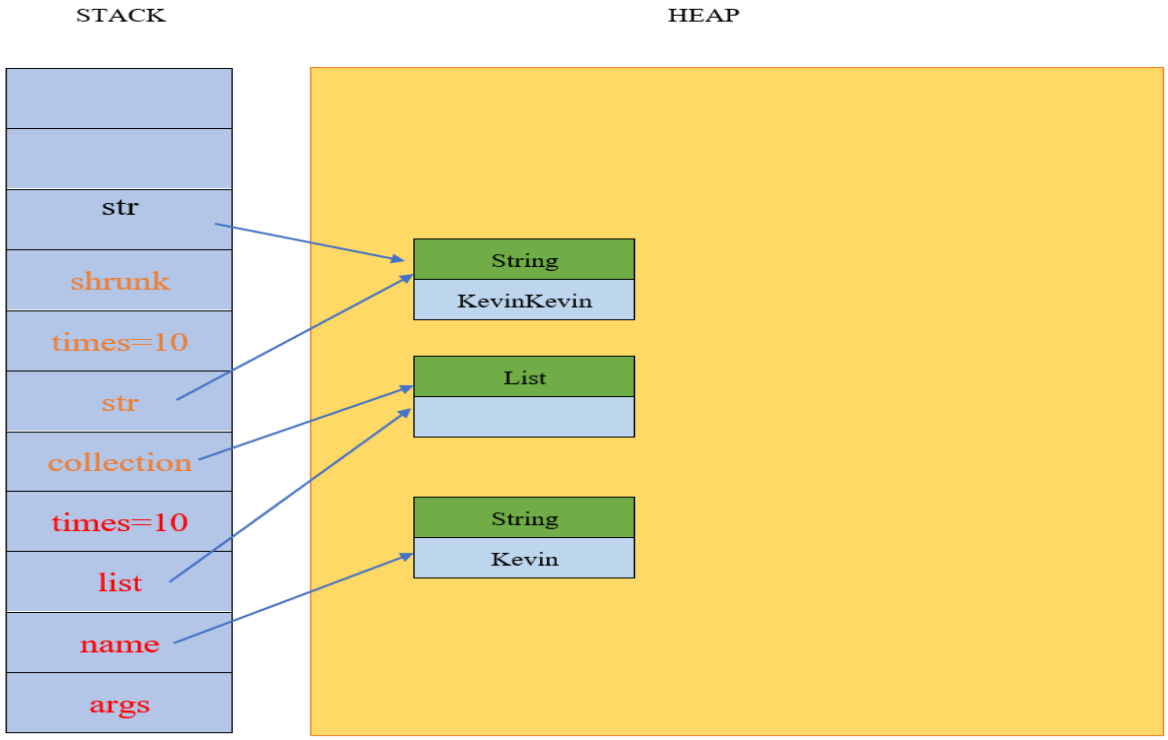
System.out.println (times + fill(list, name + name, times));

public static int fill(Collection<String> collection, String str, int times){



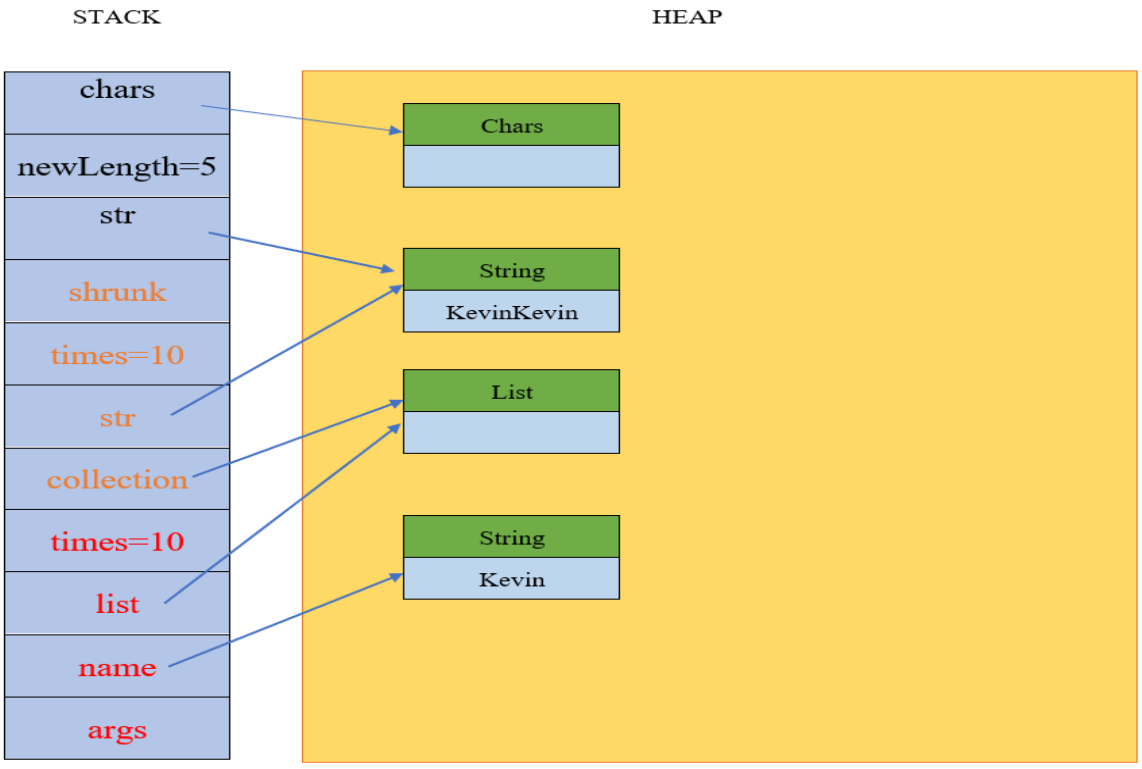
String shrunk = shrink(str);

public static String shrink(String str){

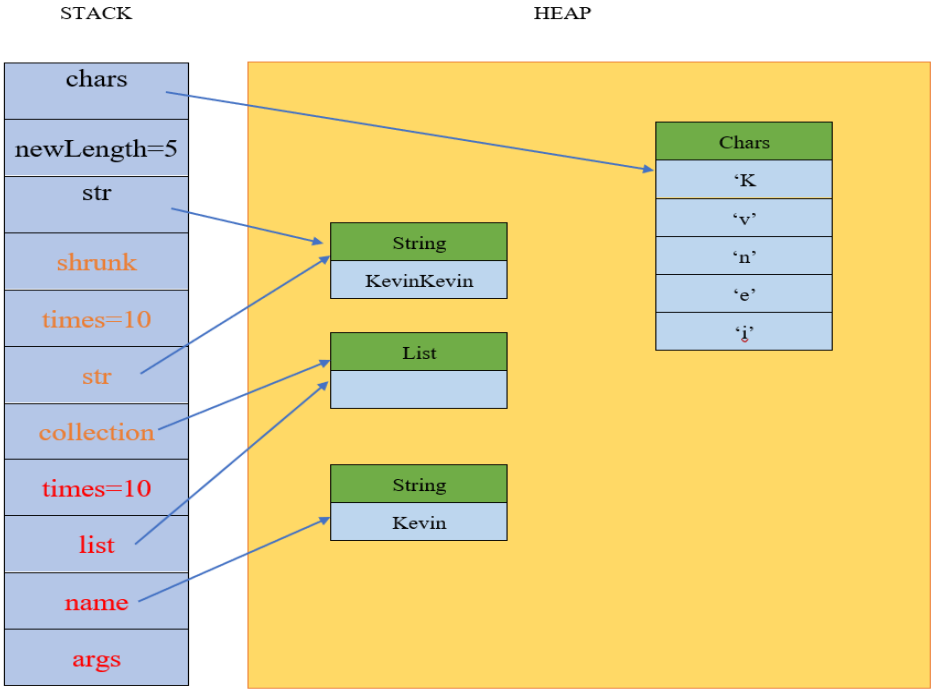


int newLength = str.length() / 2 + str.length() % 2;

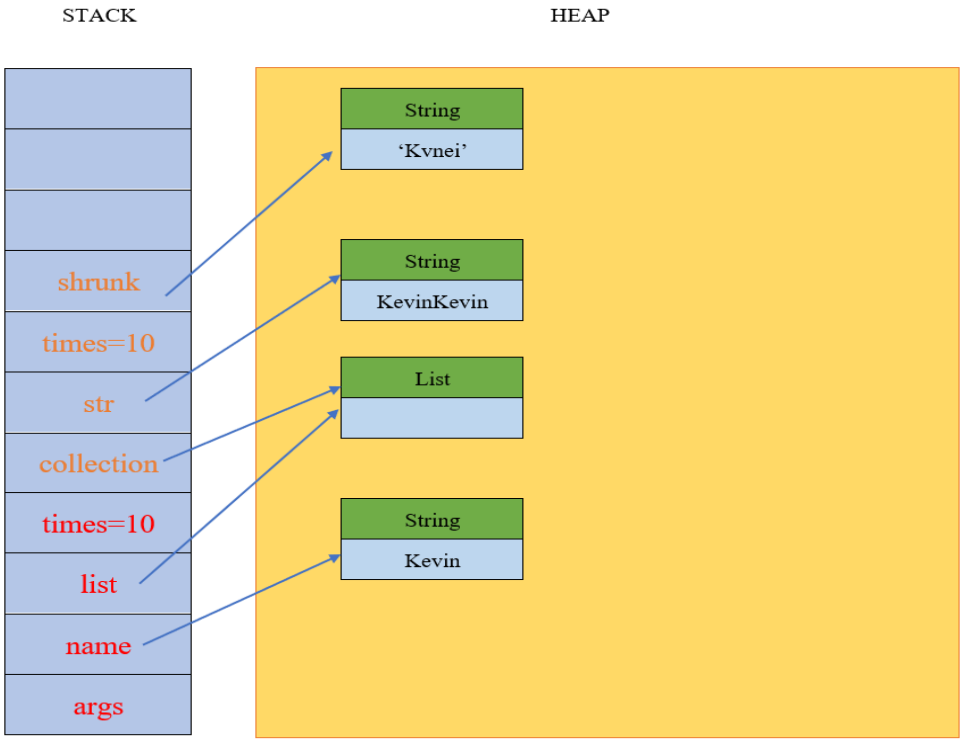
char[] chars = new char[newLength];



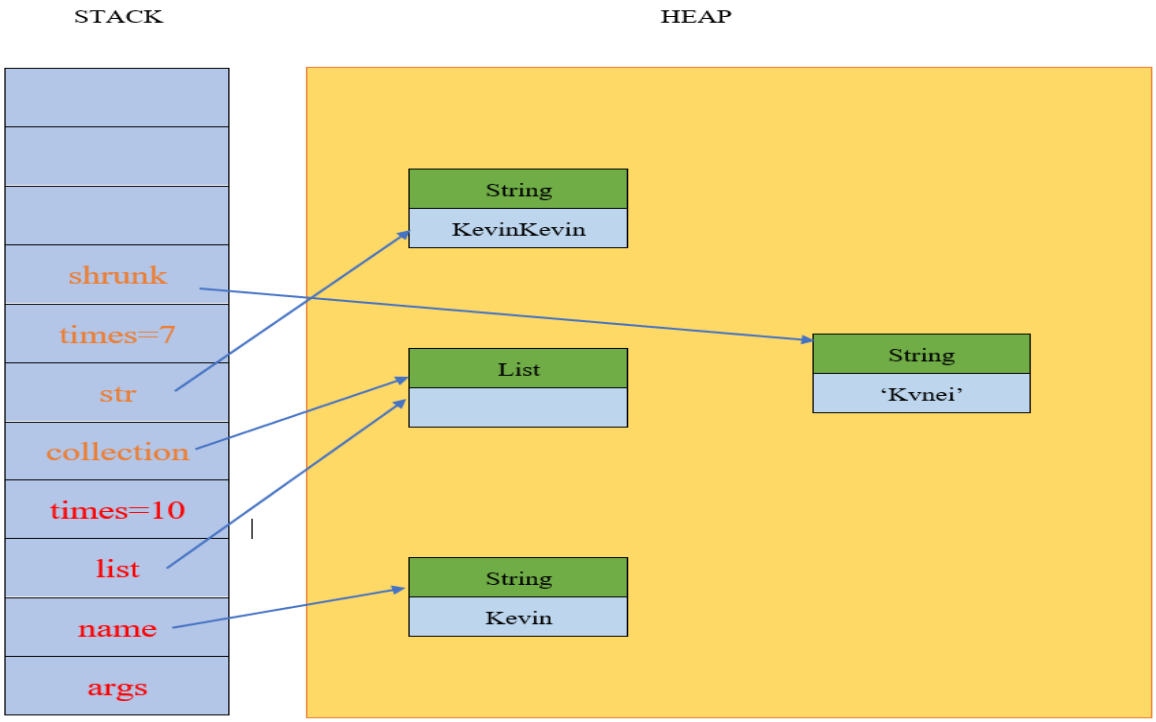
```
for (int i = 0; i < str.length(); i+=2) {
        chars[i / 2] = str.charAt(i);
    }
```



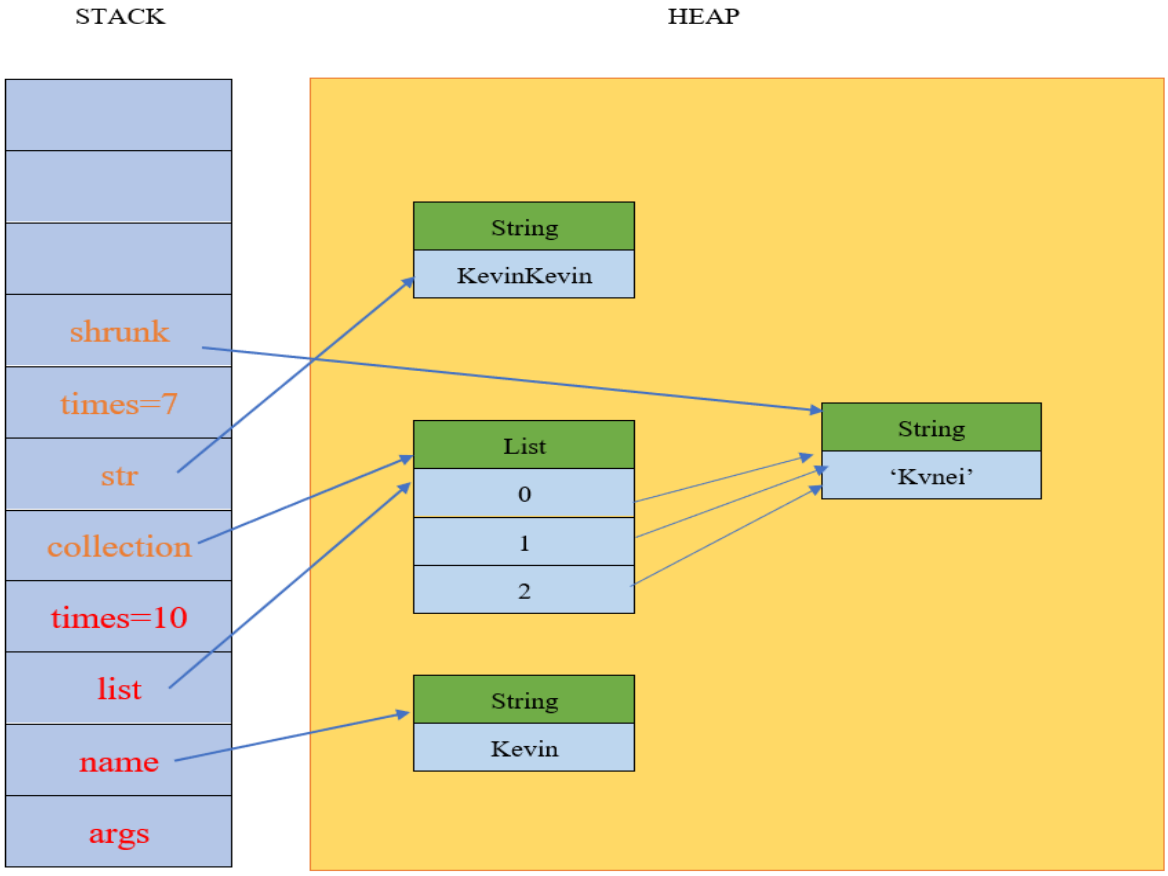

```
    return new String(chars); }
String shrunk = shrink(str);
```



```
times = (times + shrunk.length()) / 2;
```




```
for (int i = 0; i < times / 2; i++) {
        collection.add(shrunk);
```



```
return times;
System.out.println (times + fill(list, name + name, times));
    }
```

Output: 17