

Antes de refactorizar:

```
package sinLSP;

public abstract class Member {
    private final String nombre;

    public Member(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }

    public abstract void joinTournament();
    public abstract void organizeTournament();
}

package sinLSP;

public class PremiumMember extends Member {
    public PremiumMember(String nombre) {
        super(nombre);
    }
    @Override
    public void joinTournament() {
        System.out.printf("Miembro %s se une al torneo\n", getNombre());
    }
    @Override
    public void organizeTournament() {
        System.out.printf("Miembro %s organiza el torneo\n", getNombre());
    }
}

package sinLSP;

public class VipMember extends Member {
    public VipMember(String nombre) {
        super(nombre);
    }

    @Override
    public void joinTournament() {
        System.out.printf("Miembro %s se une al torneo\n", getNombre());
    }

    @Override
    public void organizeTournament() {
        System.out.printf("Miembro %s organiza el torneo\n", getNombre());
    }
}

package sinLSP;

public class FreeMember extends Member {
    public FreeMember(String nombre) {
        super(nombre);
    }
    @Override
    public void joinTournament() {
        System.out.printf("Miembro %s se une al torneo\n", getNombre());
    }

    // No es compatible con LSP:
    @Override
    public void organizeTournament() {
        System.out.printf("Miembro %s no puede organizar torneos!\n",
getNombre());
    }
}
```

```

package sinLSP;

import java.util.List;

public class Main {
    public static void main(String[] args) {
        List<Member> miembros = List.of(
            new PremiumMember("Abejita Azul"),
            new VipMember("Kaperucita Feliz"),
            new FreeMember("Inspectora Motita")
        );
        for(Member miembro : miembros){
            miembro.joinTournament();
            miembro.organizeTournament(); // No es compatible con LSP
        }
    }
}

```

Luego de refactorizar:

```

package conLSP;

public abstract class Member {
    private final String nombre;

    public Member(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }

    public abstract void joinTournament();
}

```

```

package conLSP;

public interface TopMember {
    public void organizeTournament();
}

```

```

package conLSP;

public class PremiumMember extends Member implements TopMember{
    public PremiumMember(String nombre) {
        super(nombre);
    }
    @Override
    public void joinTournament() {
        System.out.printf("Miembro %s se une al torneo\n", getNombre());
    }
    @Override
    public void organizeTournament() {
        System.out.printf("Miembro %s organiza el torneo\n", getNombre());
    }
}

```

```

package conLSP;

public class VipMember extends Member implements TopMember{
    public VipMember(String nombre) {
        super(nombre);
    }
    @Override

```

```

    public void joinTournament() {
        System.out.printf("Miembro %s se une al torneo\n", getNombre());
    }
    @Override
    public void organizeTournament() {
        System.out.printf("Miembro %s organiza el torneo\n", getNombre());
    }
}

```

```

package conLSP;

public class FreeMember extends Member {
    public FreeMember(String nombre) {
        super(nombre);
    }
    @Override
    public void joinTournament() {
        System.out.printf("Miembro %s se une al torneo\n", getNombre());
    }
}

```

```

package conLSP;

import java.util.List;

public class Main {
    public static void main(String[] args) {
        List<TopMember> miembrosTop = List.of(
            new PremiumMember("Abejita Azul"),
            new VipMember("Kaperucita Feliz")
        );
        for(TopMember miembro : miembrosTop){
            miembro.organizeTournament();
        }

        List<Member> miembros = List.of(
            new PremiumMember("Abejita Azul"),
            new VipMember("Kaperucita Feliz"),
            new FreeMember("Inspectora Motita")
        );
        for(Member miembro : miembros){
            miembro.joinTournament();
        }
    }
}

```