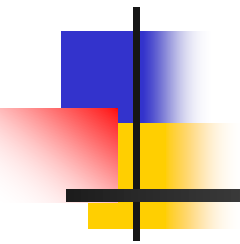
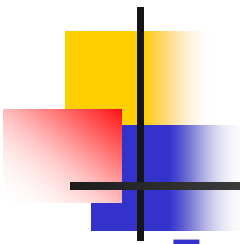


Construcción de parsers



Ejemplo de lenguaje

- Supongamos que el usuario de una aplicación puede escribir comandos como los siguientes:
  - help
  - quit
  - reset
  - replace 6 (o replace 8, replace 12, ...)
  - run
- Vamos a ver cómo analizar estos strings con un parser.

La clase CommandParser

```
public class CommandParser {
    private final static Command[] commands=
      {new Help(), new Quit(), new Reset(),
       new Run(), new Replace(0)};

    public static Command parseCommand(String line, …) {
      //Quitar blancos y dividir en tokens
      if (tokens.length==0 || tokens.length>2) ...
      Command cm;
      for (Command c:commands) {
           cm=c.parse(tokens);
           if (cm!=null) return cm;
      }
```

La clase CommandParser

```
    public static String showHelp() {
        String s="";
        for (Command c:commands) s+=c.helpText();
        return s;
    }
}
```

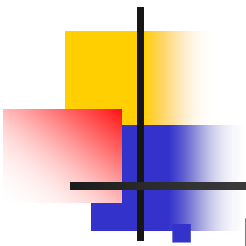
La clase abstracta **Command**

```
public abstract class Command {
    ...
    public abstract void execute(...);
    public abstract Command parse(String[] tokens, …);
    public String helpText() {...}
}
```

La clase **Reset**

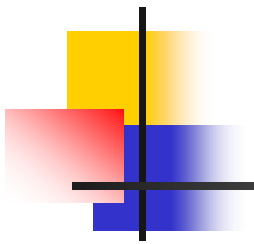
```
public class Reset extends Command {
    ...
    public void execute(Game g, ...) {
       g.executeReset();
    }

    public Command parse(String[] s) {
       if (s.length==1 &&
           s[0].equalsIgnoreCase("RESET"))
           return new Reset();
       else return null;
    }
    ...
}
```



La clase `Replace`

```
public class Replace extends Command {
    private int n;
    ...
    public void execute(Game g, ...) {
       g.executeReplace();
    }
```



La clase Replace

```
    public Command parse(String[] s) {
        if (s.length!=2) return null;
        if (!s[0].equalsIgnoreCase("REPLACE")) return null;

        //Escaneamos s[1] en una variable m

        if (m no es entero) return null;
        else return new Replace(m);
    }
    ...
}
```

La clase **Controller**

```
public void run() {
    while (!game.isFinished()) {
        ...
        System.out.print("Command > ");
        String[] words =
            scanner.nextLine().toLowerCase().
                trim().split(“ s+”);
        Command command =
            CommandParser.parseCommand(words, this);
        if (command != null)
            command.execute(game, this);
        else ...
```