Object-oriented programming

Second semester

Lecture №8

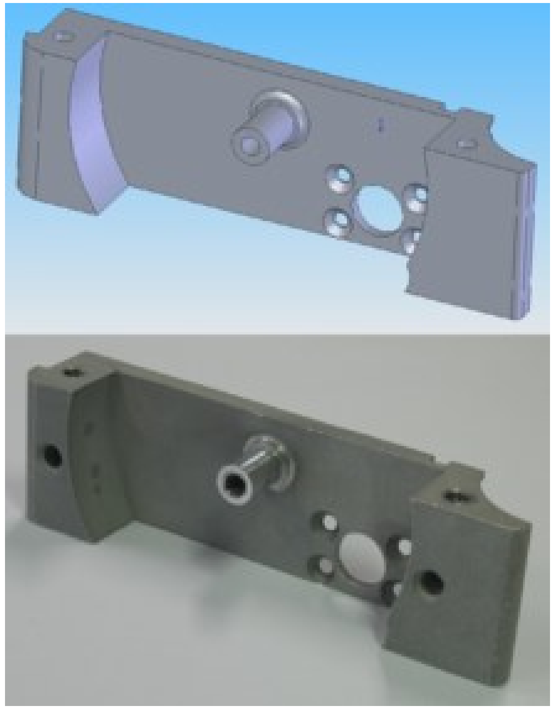
DSL

Domain-Specific Language (DSL)

A domain-specific language (DSL) is a computer language specialized to a particular application domain. This is in contrast to a general-purpose language (GPL), which is broadly applicable across domains.

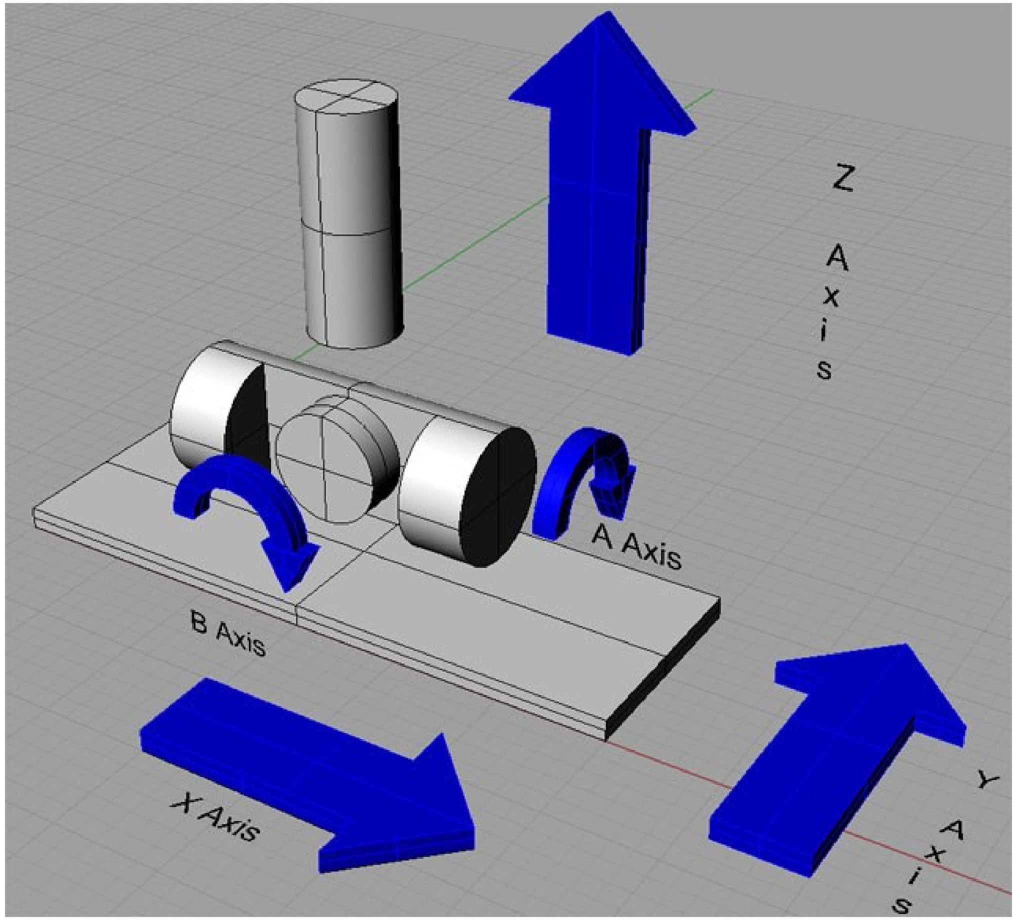
- TeX/LaTeX
- SQL
- HTML
- AutoLisp
- Prolog
- G-Code

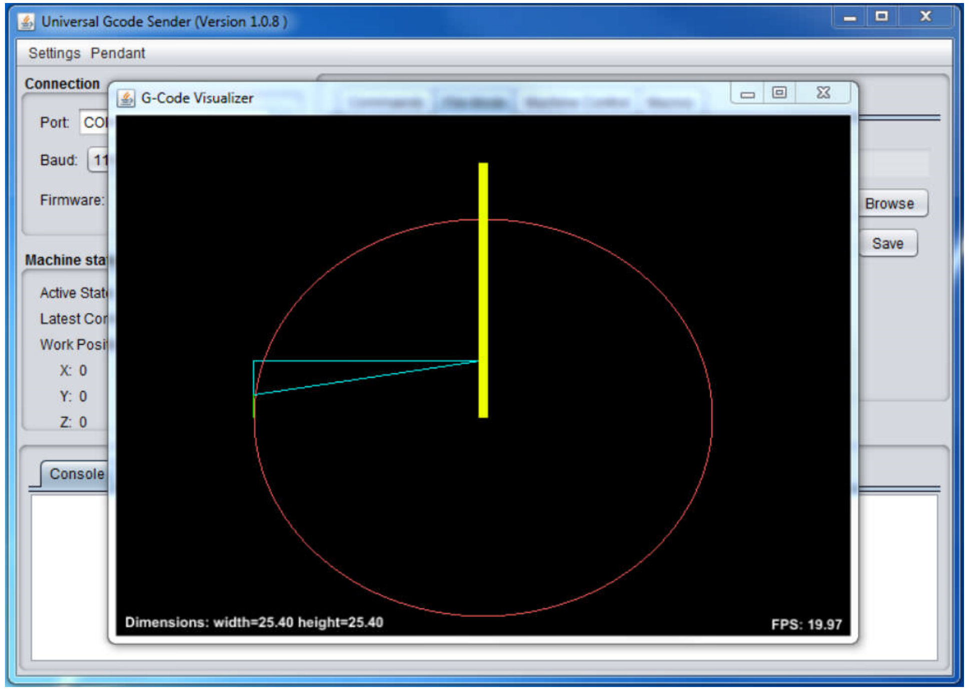
G-Code



G-Code

```
G17 G20 G90 G94 G54
G0 Z0,25
X-0,5 Y0.
Z0,1
G01 Z0. F5.
G02 X0. Y0,5 I0,5 J0. F2,5
X0,5 Y0. I0. J-0,5
X0. Y-0,5 I-0,5 J0.
X-0,5 Y0. I0. J0,5
G01 Z0,1 F5.
G00 X0. Y0. Z0.25
```





Port: CO

Baud: 11

Firmware:

Machine sta

Active Stat

Latest Cor

Work Posi

X: 0

Y: 0

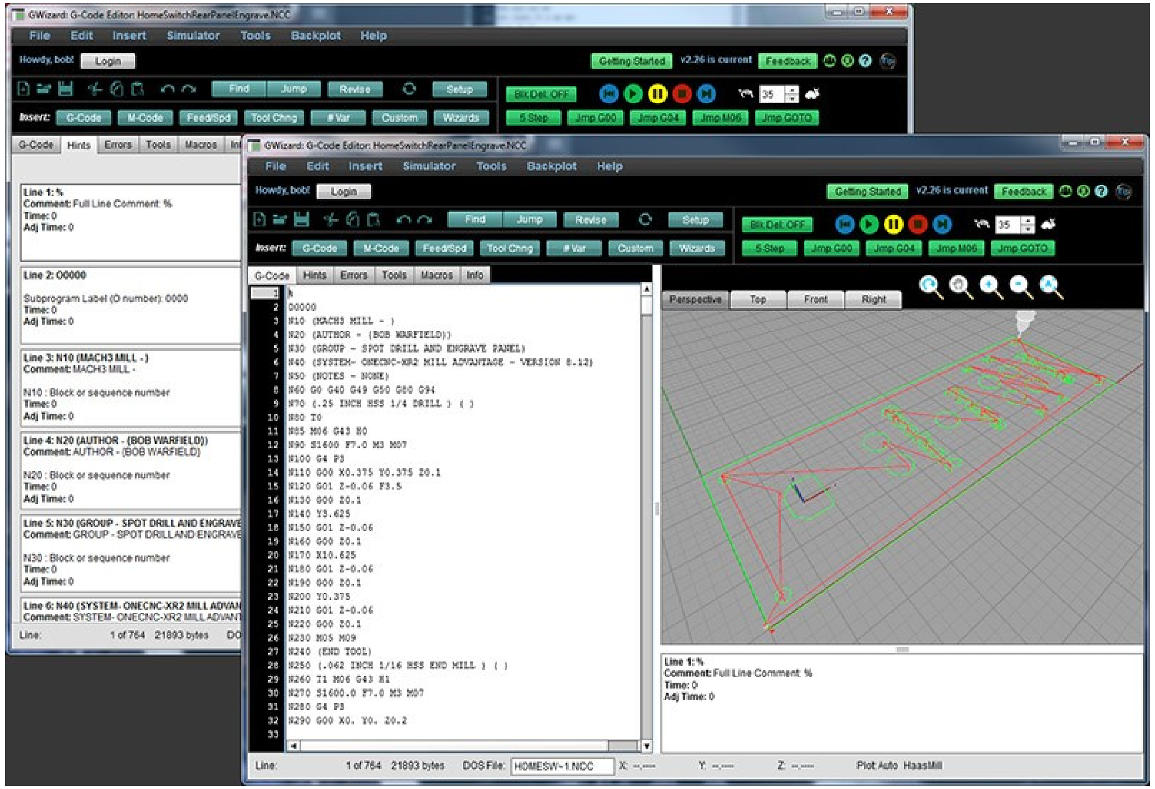
Z: 0

Dimensions: width=25.40 height=25.40

FPS: 19.97

This simple program will draw a 1” diameter circle about the origin.

G-Code



Line 1: %
Comment: Full Line Comment: %
Time: 0
Adj Time: 0

Line 2: O0000

Subprogram Label (O number): 0000
Time: 0
Adj Time: 0

Line 3: N10 (MACH3 MILL -)
Comment: MACH3 MILL -

N10 : Block or sequence number
Time: 0
Adj Time: 0

Line 4: N20 (AUTHOR - (BOB WARFIELD))
Comment: AUTHOR - (BOB WARFIELD)

N20 : Block or sequence number
Time: 0
Adj Time: 0

Line 5: N30 (GROUP - SPOT DRILL AND ENGRAVE
Comment: GROUP - SPOT DRILL AND ENGRAVE

N30 : Block or sequence number
Time: 0
Adj Time: 0

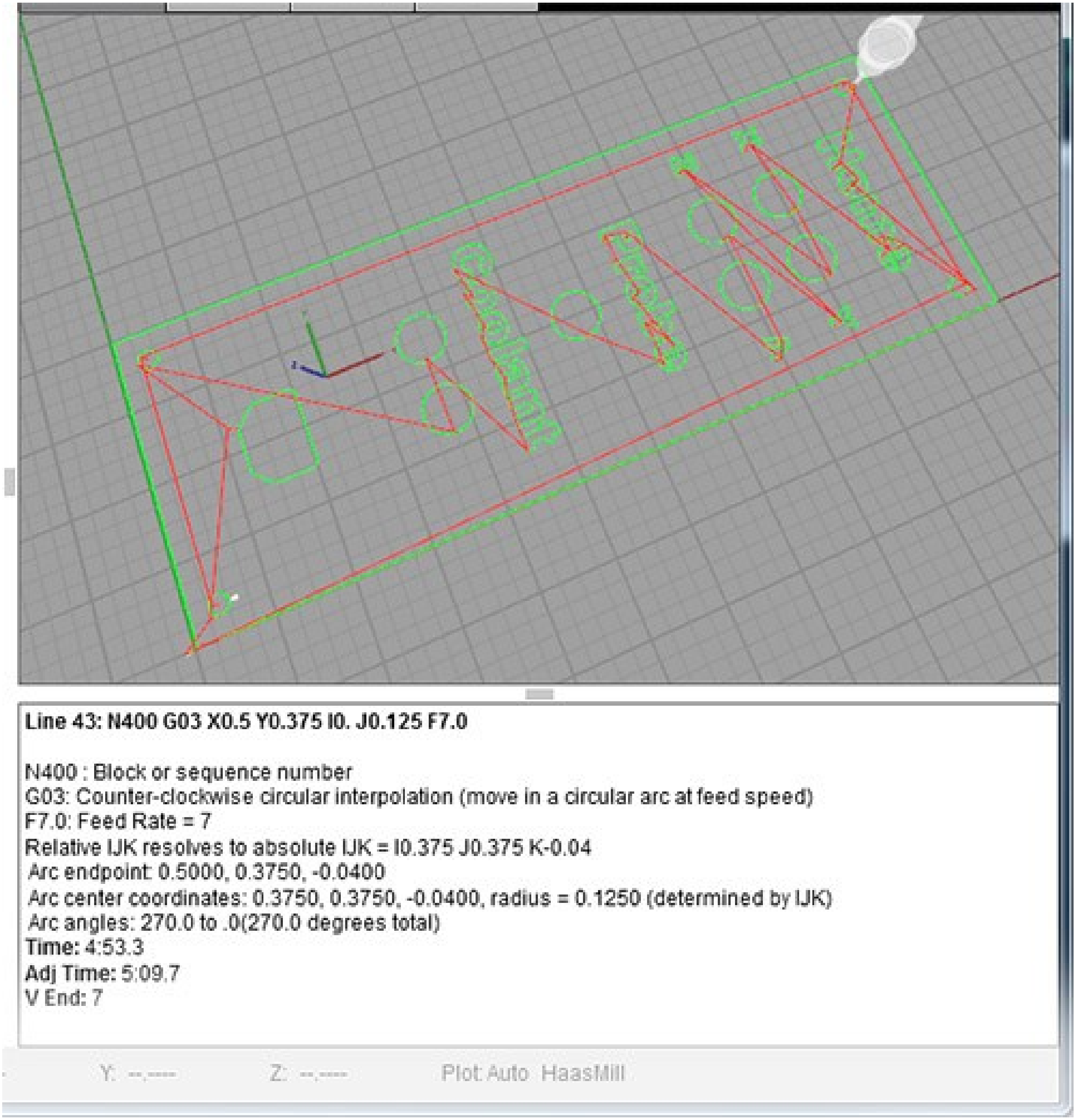
Line 6: N40 (SYSTEM- ONECNC-XR2 MILL ADVAN
Comment: SYSTEM- ONECNC-XR2 MILL ADVANT

Line: 1 of 764 21893 bytes

```
%
O0000
N10 (MACH3 MILL - )
N20 (AUTHOR - (BOB WARFIELD))
N30 (GROUP - SPOT DRILL AND ENGRAVE PANEL)
N40 (SYSTEM- ONECNC-XR2 MILL ADVANTAGE - VERSION 8.12)
N50 (NOTES - NONE)
N60 G0 G40 G49 G50 G80 G94
N70 (.25 INCH HSS 1/4 DRILL ) ( )
N80 T0
N85 M06 G43 H0
N90 S1600 F7.0 M3 M07
N100 G4 P3
N110 G00 X0.375 Y0.375 Z0.1
N120 G01 Z-0.06 F3.5
N130 G00 Z0.1
N140 Y3.625
N150 G01 Z-0.06
N160 G00 Z0.1
N170 X10.625
N180 G01 Z-0.06
N190 G00 Z0.1
N200 Y0.375
N210 G01 Z-0.06
N220 G00 Z0.1
N230 M05 M09
N240 (END TOOL)
N250 (.062 INCH 1/16 HSS END MILL ) ( )
N260 T1 M06 G43 H1
N270 S1600.0 F7.0 M3 M07
N280 G4 P3
N290 G00 X0. Y0. Z0.2
```

Line 1: %
Comment: Full Line Comment: %
Time: 0
Adj Time: 0

Line: 1 of 764 21893 bytes DOS File: HOMESW~1.NCC X: --.--- Y: --.--- Z: --.--- Plot: Auto HaasMill

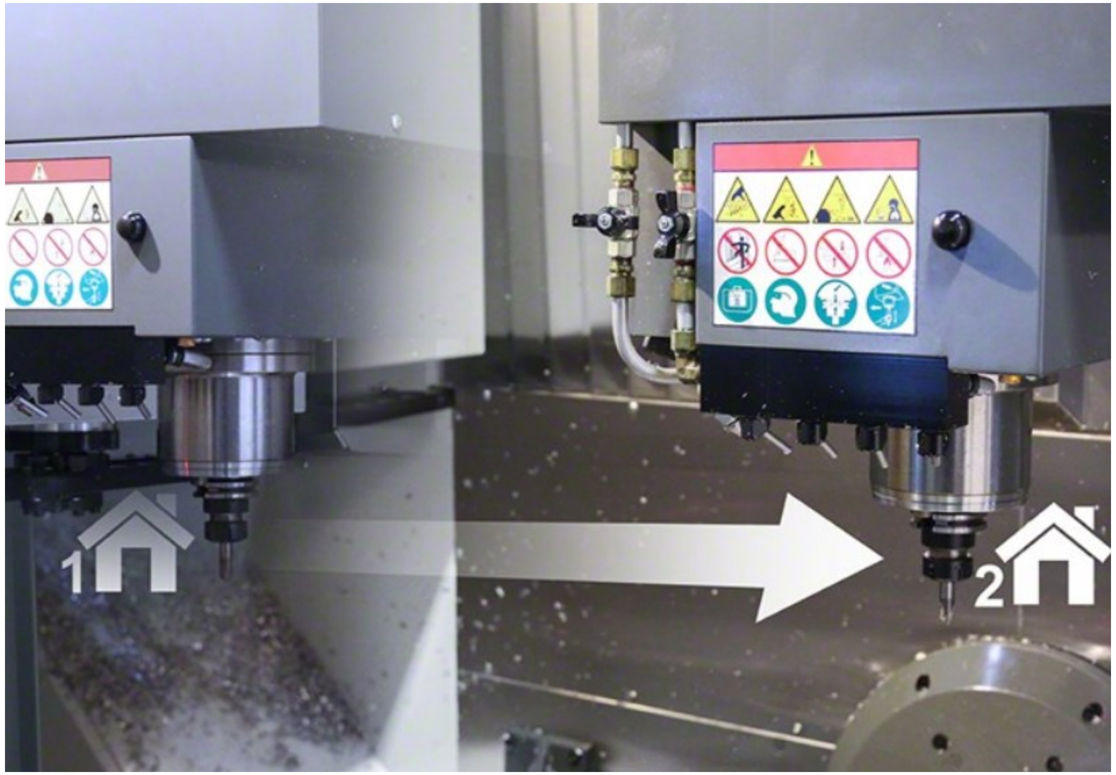


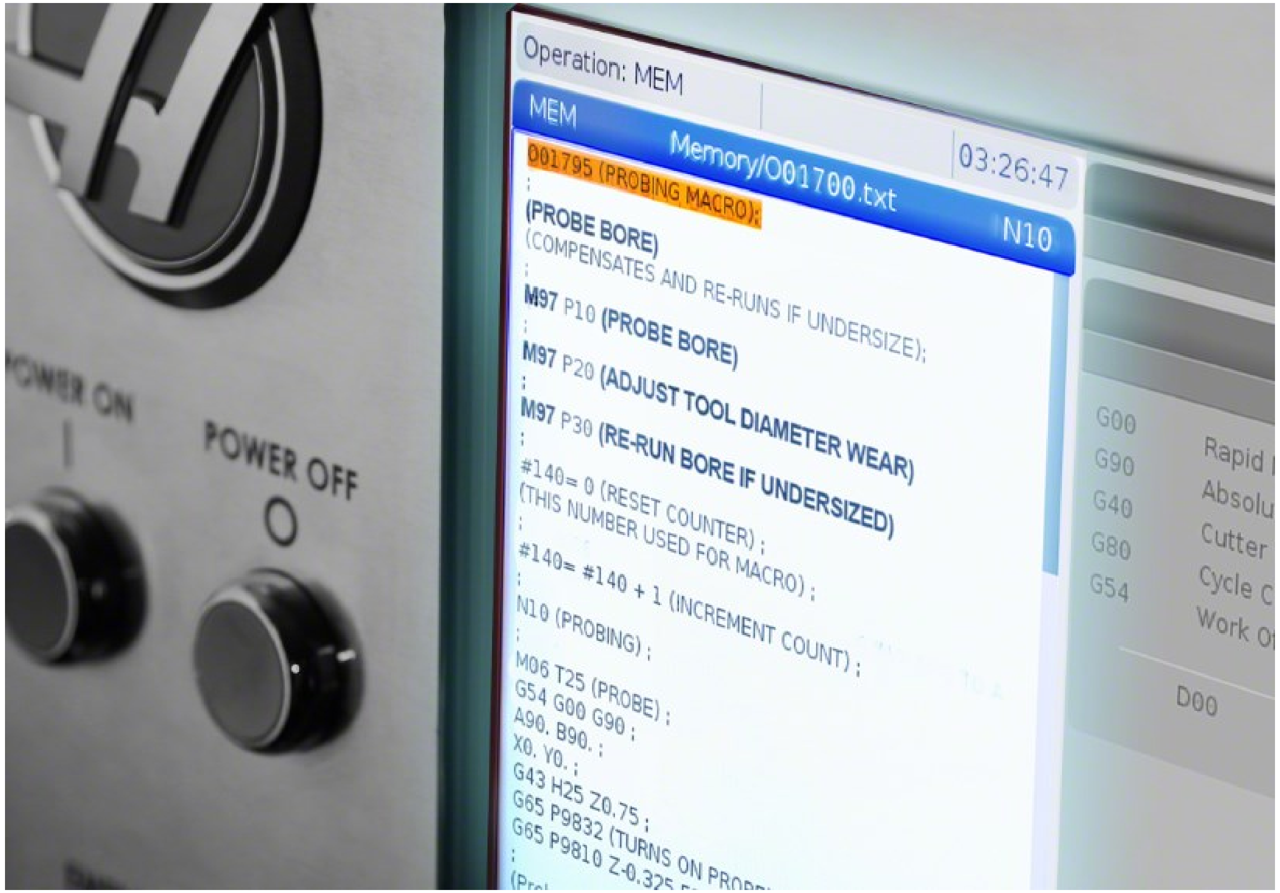
Line 43: N400 G03 X0.5 Y0.375 I0. J0.125 F7.0

N400 : Block or sequence number
G03: Counter-clockwise circular interpolation (move in a circular arc at feed speed)
F7.0: Feed Rate = 7
Relative IJK resolves to absolute IJK = I0.375 J0.375 K-0.04
Arc endpoint: 0.5000, 0.3750, -0.0400
Arc center coordinates: 0.3750, 0.3750, -0.0400, radius = 0.1250 (determined by IJK)
Arc angles: 270.0 to .0(270.0 degrees total)
Time: 4:53.3
Adj Time: 5:09.7
V End: 7

Y: --.--- Z: --.--- Plot: Auto HaasMill

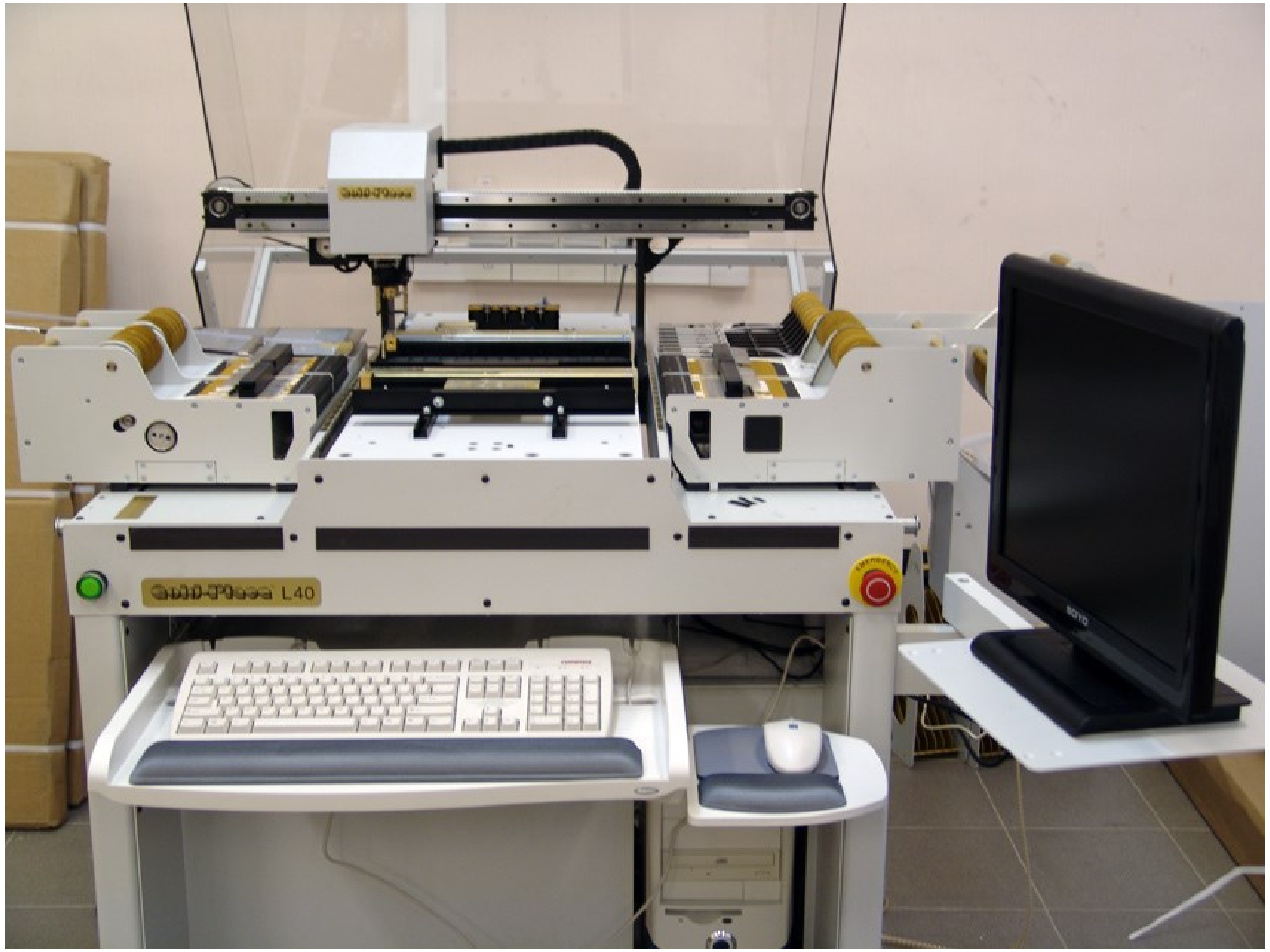
G-Code



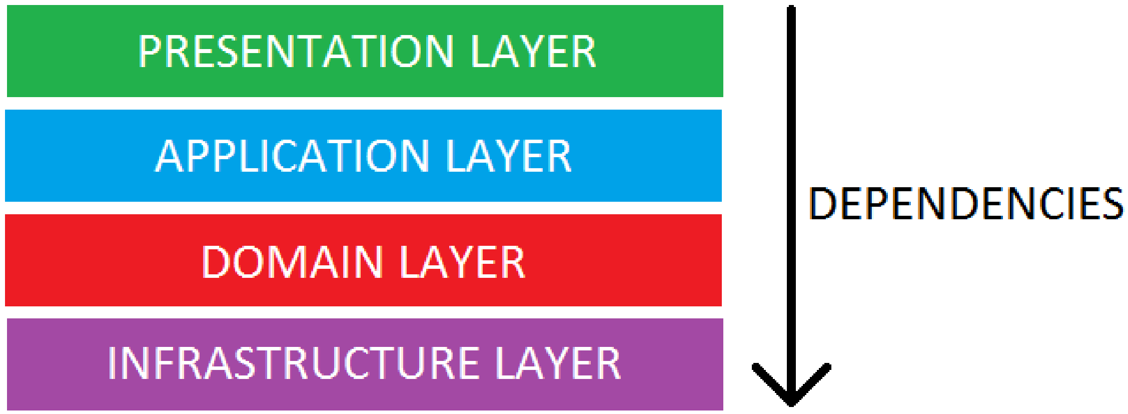


```
001795 (PROBING MACRO);
;
(PROBE BORE)
(COMPENSATES AND RE-RUNS IF UNDERSIZE);
;
M97 P10 (PROBE BORE)
;
M97 P20 (ADJUST TOOL DIAMETER WEAR)
;
M97 P30 (RE-RUN BORE IF UNDERSIZED)
;
#140= 0 (RESET COUNTER) ;
(THIS NUMBER USED FOR MACRO) ;
;
#140= #140 + 1 (INCREMENT COUNT) ;
;
N10 (PROBING) ;
;
M06 T25 (PROBE) ;
G54 G00 G90 ;
A90. B90. ;
X0. Y0. ;
G43 H25 Z0.75 ;
G65 P9832 (TURNS ON PROBE
G65 P9810 Z-0.325
;
(Pro
```

Printboard



DSL & OOP, Multilayered architecture



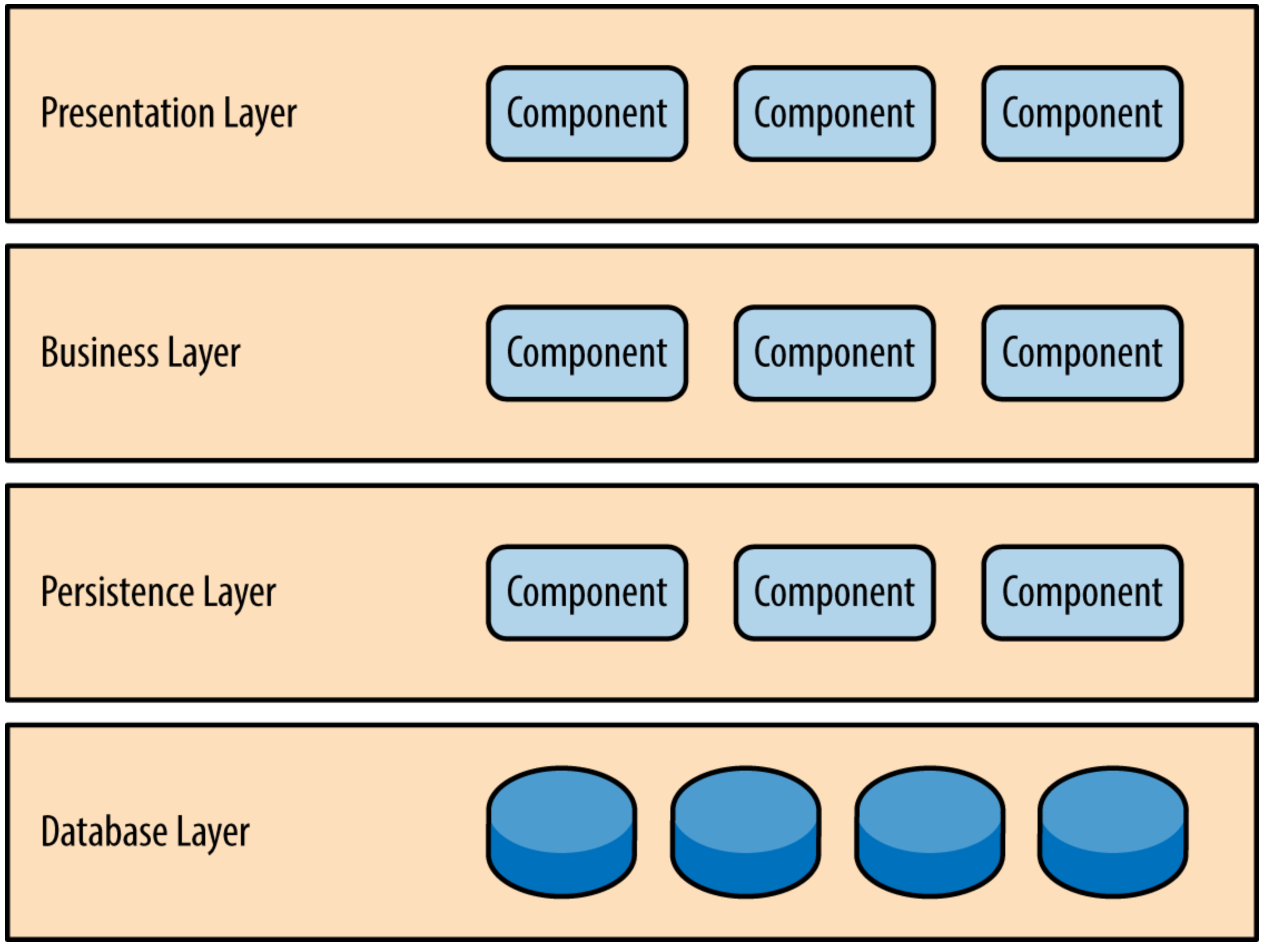
PRESENTATION LAYER

APPLICATION LAYER

DOMAIN LAYER

INFRASTRUCTURE LAYER

DEPENDENCIES



Presentation Layer

Component

Component

Component

Business Layer

Component

Component

Component

Persistence Layer

Component

Component

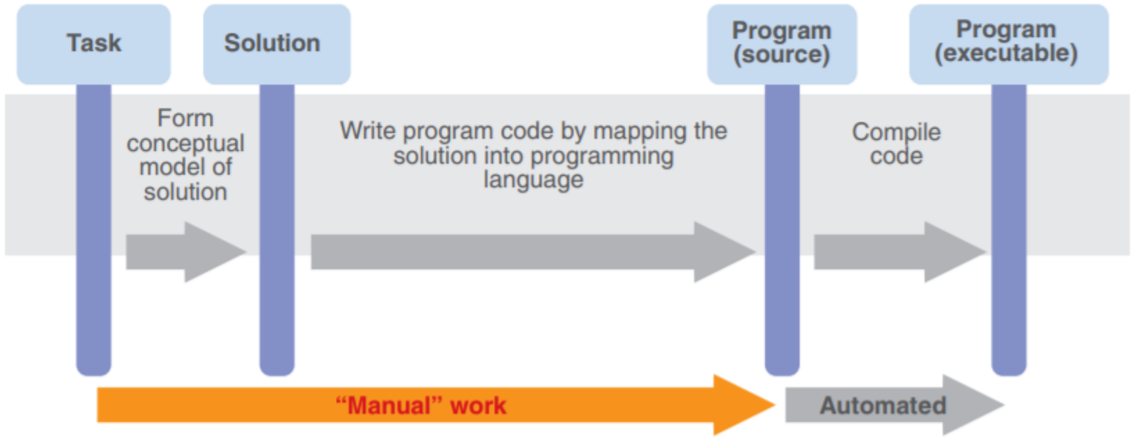
Component

Database Layer

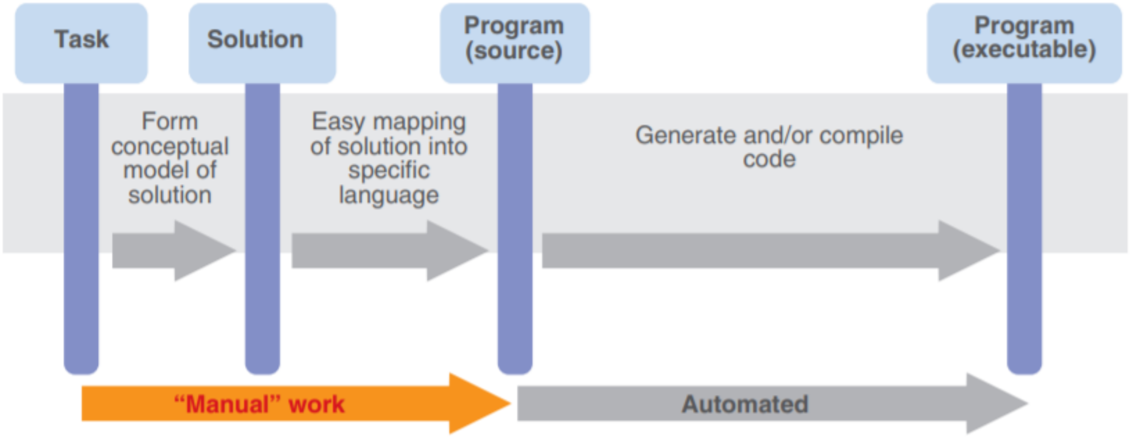
DSL

- Environment objects
- Instance of objects
- Domain’s types
- Domain’s rules
- Domain’s algorithms ?
  (Who knows how to solve the problem of the business)

General-purpose and domain-specific languages


Task
Solution
Program (source)
Program (executable)
Form conceptual model of solution
Write program code by mapping the solution into programming language
Compile code
"Manual" work
Automated

Mainstream programming with a general-purpose language.


Task
Solution
Program (source)
Program (executable)
Form conceptual model of solution
Easy mapping of solution into specific language
Generate and/or compile code
"Manual" work
Automated

Language-oriented programming with domain-specific languages.

Defining a new DSL

With this kind of setup in place, there are three main parts to defining a new DSL:

- Define the abstract syntax, that is the **schema** of the abstract representation.
- Define an **editor** to let people manipulate the abstract representation through a projection.
- Define a **generator**. This describes how to translate the abstract representation into an executable representation. In practice the generator defines the semantics of the DSL.

```
mustRunAfter "taskX"
```

Building Java Projects with Gradle

```
apply plugin: 'java'
apply plugin: 'eclipse'
apply plugin: 'application'

mainClassName = 'hello.HelloWorld'

// tag::repositories[]
repositories {
   mavenCentral()
}
// end::repositories[]

// tag::jar[]
jar {
   baseName = 'gs-gradle'
   version =  '0.1.0'
}
// end::jar[]
```

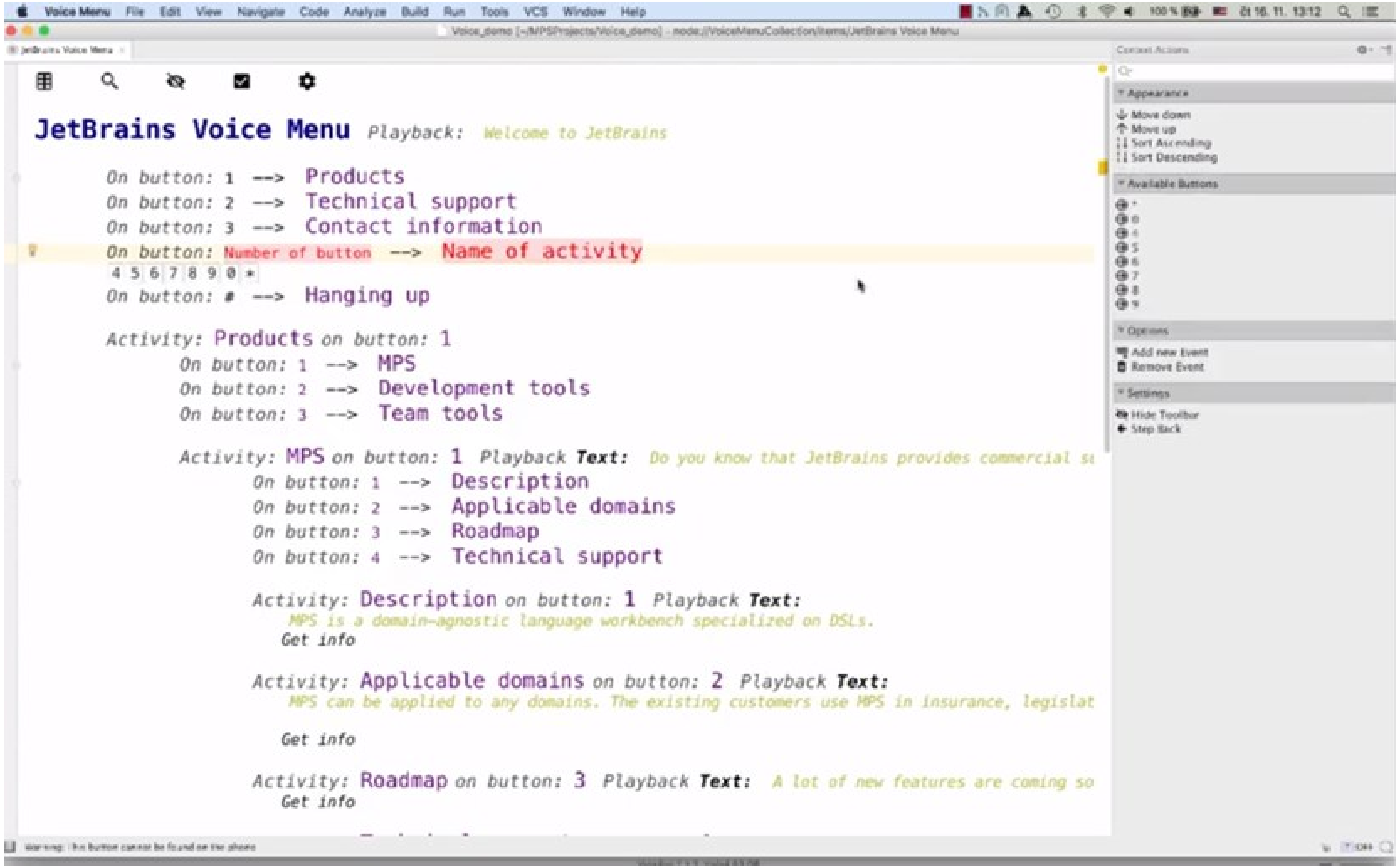
```
// tag::dependencies[]
sourceCompatibility = 1.8
targetCompatibility = 1.8

dependencies {
   compile "joda-time:joda-time:2.2"
   testCompile "junit:junit:4.12"
}
// end::dependencies[]

// tag::wrapper[]
// end::wrapper[]

task myJavadocs(type: Javadoc) {
   timeout = Duration.ofMinutes(10)
   source = sourceSets.main.allJava
   mustRunAfter "taskX"
}
```

DSL example



```
JetBrains Voice Menu Playback: Welcome to JetBrains

        On button: 1 --> Products
        On button: 2 --> Technical support
        On button: 3 --> Contact information
        On button: Number of button --> Name of activity
        4 5 6 7 8 9 0 *
        On button: # --> Hanging up

        Activity: Products on button: 1
                On button: 1 --> MPS
                On button: 2 --> Development tools
                On button: 3 --> Team tools

                Activity: MPS on button: 1 Playback Text: Do you know that JetBrains provides commercial su
                        On button: 1 --> Description
                        On button: 2 --> Applicable domains
                        On button: 3 --> Roadmap
                        On button: 4 --> Technical support

                        Activity: Description on button: 1 Playback Text:
                            MPS is a domain-agnostic language workbench specialized on DSLs.
                           Get info

                        Activity: Applicable domains on button: 2 Playback Text:
                            MPS can be applied to any domains. The existing customers use MPS in insurance, legislat

                           Get info

                        Activity: Roadmap on button: 3 Playback Text: A lot of new features are coming so
                           Get info
```

Warning: This button cannot be found on the phone