Variability Implementation: Parameters

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Parameters

- There are many ways to implement variable code
  - A simple way is the use of conditional statements, such as *if* and *switch*
- Conditional statements are typically controlled by parameters
  - Different parameters lead to different executions

Defining and Storing Values

- Values of parameters can be defined in different ways
  - Command line
  - Configuration file
- Global variables are a common way to store and access values of parameters
  - Usually one Boolean variable per feature

Example of Code: Configuration

```
public class Configuration {
  public static boolean SORTING = true;
  public static boolean FAVOURITES = true;
  ...
}
```

- Global variables are a common way to store and access values of parameters
  - Usually one Boolean variable per feature

Example of Code: PhotoListScreen

```
public class PhotoListScreen extends List {
  ...
  public void initMenu() {
    this.addCommand(viewCommand);
    this.addCommand(addCommand);
    this.addCommand(deleteCommand);
    this.addCommand(editLabelCommand);

    if (Configuration.SORTING) {
      this.addCommand(sortCommand);
    }

    if (Configuration.FAVOURITES) {
      this.addCommand(favoriteCommand);
      this.addCommand(viewFavoritesCommand);
    }
  }
}
```

Example of Code: PhotoListScreen

```
public class PhotoListScreen extends List {
  ...
  public void initMenu() {
    this.addCommand(viewCommand);
    this.addCommand(addCommand);
    this.addCommand(deleteCommand);
    this.addCommand(editLabelCommand);
    if (Configuration.SORTING) {
      this.addCommand(sortCommand);
    }
    if (Configuration.FAVOURITES) {
      this.addCommand(favoriteCommand);
      this.addCommand(viewFavoritesCommand);
    }
  }
}
```

S

F

Use in Practice

- Implementing variability with parameters is easy
  - It is widely used in practice

- All code is deployed in every product
  - It may impact on resource consumption, performance, and security
  - Compilers can do some optimizations, e.g., remove dead code of unused features

Parameter is Annotation

- Conditional statements are a form of annotation
  - They allow fine granularity, e.g., statement level and expression level (a ? b : c)
- They can essentially used in all programming languages

- They are not applicable to non-code artifacts, such as documentation

Poor Code Quality

- The use of parameters can lead to poor code quality
  - Global variables reduce modularity
  - Conditional statements make code hard to be understood

- Feature code is usually scattered and tangled across multiple files

Summary of Benefits

- Easy to use and well-known
- Flexible and fine grained
- Allow runtime (re)configuration

Summary of Drawbacks

- All code is deployed
  - No compile time configuration
- Often used in undisciplined fashion
- Code scattering and tangling
- Extension requires invasive changes
- No support for non-code artifacts

Bibliography

- S. Apel, D. Batory, C. Kastner, G. Saake. **Feature-Oriented Software Product Lines: Concepts and Implementation**. Springer; 2013.
  - Section 4.1