

Dependency Breaking Kata

Week 8: Practical Session

Advanced Software Engineering

Learning Objectives

- Practice breaking dependencies using adapters and seams
- Refactor legacy code for testability
- Safely introduce new interfaces

Kata Instructions

Work in pairs to break dependencies in the following code.

C++ Starting Point

```
1  class EmailSender {
2  public:
3      void send(const std::string& to, const std::string& body) {
4          // Directly uses SMTP library
5          SmtpClient smtp;
6          smtp.connect("smtp.company.com");
7          smtp.sendMail(to, body);
8          smtp.disconnect();
9      }
10 };
```

Python Starting Point

```
1  class EmailSender:
2      def send(self, to, body):
3          # Directly uses SMTP library
4          smtp = SmtpClient()
5          smtp.connect("smtp.company.com")
6          smtp.send_mail(to, body)
```

Tasks

1. **Identify Hard Dependencies (15 min)**
 - List all direct dependencies in the code
 - Discuss why they make testing difficult
2. **Introduce Adapters (25 min)**
 - Refactor to use an interface or protocol for the SMTP client
 - Allow injection of a mock or test double
3. **Create Seams (30 min)**
 - Identify and implement seams for testing
 - Write tests using the new seams/adapters
4. **Review and Discuss (20 min)**
 - Share your refactoring and tests with another pair
 - Discuss trade-offs and risks

Success Criteria

- Dependencies broken via adapters or seams
- Code is testable with mocks or stubs
- Minimal changes to legacy logic
- Risks and trade-offs discussed

Notes for Facilitators

- Encourage use of interfaces/protocols
- Focus on testability and minimal change
- Discuss real-world applications of these techniques

Follow-up

Prepare for next week's session on:

- Large-scale refactoring
- Managing technical debt
- Strategic improvement planning