**강 의 계 획 서**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **교 과**  **목 명** | 고급객체지향자바개발론 | | | **학점/**  **시수** | 3 |
| **담 당**  **교 수** | 백 명석 | **E-mail** | msbaek@daumcorp.com | **연구실 전화** |  |

**1. 과목 개요**

한번 작성된 코드는 이해하거나 수정하기 위해 1,000 이상 읽혀진다고 한다. 또 돌아가는 코드는 누구나 작성할 수 있지만 읽기 좋은 코드는 훌륭한 아키텍트만이 작성할 수 있다고 한다. Daum에서의 서비스는 한번 만들어지면 짧게는 수년에서 길게는 10년을 넘게 지속적으로 버그를 수정하고, 새로운 요구사항을 반영하며 운영된다. 본 강의에서는 객체지향언어인 자바를 활용하여 유지보수 가능한 프로그램 개발을 위한 방법들을 소개하고 실습을 통하여 익힐 수 있도록 한다.

본 강의에서는 객체지향의 기본 개념, TDD(Test Driven Development), Architecture, SOLID Principles 등에 대해서 학습하고, 다양한 플러그인을 포함한 Eclipse 환경에서 학습한 내용을 실습을 통해 익히고, 학습한 내용을 활용하여 Term 프로젝트를 수행한다.

**2. 대상 학생**

Daum 트랙 과정 학생 (컴퓨터공학과 및 컴퓨터 교육과 3/4학년)

**3. 선수과목 또는 수강 전 필요한 지식**

* 필수 - 고급 웹 프로그래밍 실습, 오픈소스 개발방법론, 객체지향 프로그래밍
* 요구사항: 본 강의 수강자는 자바 프로그래밍 언어를 사용할 수 있어야 하며, 리눅스/유닉스 운영 체제를 사용해 본 경험이 있어야 한다.

**4. 주차별 강의 내용**

|  |  |  |
| --- | --- | --- |
| 주차 | 강의 주제 | 강의 내용 |
| 1 | OOP 소개 | 객체지향과 절차지향의 차이. OOP 기본 개념 |
| 2 | 진화적 설계 소개 | 필요한 기능을 효과적으로 구현하는 방법 소개. Up-front와 Agile의 차이점, 사전 설계와 Refactoring을 통한 설계의 차이점 |
| 3 | 개발 환경 구축 | STS 소개, 설치  STS 플러그인 설치 및 실습  Github 사용법 설명 및 실습 |
| 4 | Functions | 객체지향 관점에서 함수란 무엇인가 ? 어떤 함수가 좋은 함수인가 ? |
| 5 | Functions | 리팩토링을 통한 실습. |
| 6 | Function Structure | Switch 문장의 문제점, Tell, Don’t Ask |
| 7 | Function Structure | Stack 예제를 통한 실습 |
| 8 | TDD | 개요. |
| 9 | TDD | Tyrant를 통한 실습 |
| 10 | TDD | Bowling Game을 통한 실습 |
| 11 | Architecture | Architecture란 ? 좋은 Architecture란 ? |
| 12 | Architecture | Main Partition과 Application Partition |
| 13 | SOLID | SRP. OCP |
| 14 | SOLID | DIP |
| 15 | SOLID | ISP, LSP |
| 16 | Term Project 발표 | 프로젝트 발표 |

**5. 수업 진행 방법: 일반 강의(60%), 실습 강의(40%)**

**6. 학습 평가방법**

|  |  |
| --- | --- |
| 출석 | 50% |
| 중간고사 | 20% |
| 프로젝트 | 30% |