

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
(найменування центрального органу виконавчої влади у сфері освіти і науки)

ПРОЕКТНИЙ ПРАКТИКУМ
(назва навчальної дисципліни)

ПРОГРАМА
навчальної дисципліни
підготовки бакалаврів
(назва рівня вищої освіти (освітньо-кваліфікаційний рівень))
спеціальності 121 «Інженерія програмного забезпечення»
(код і назва спеціальності)

РОЗРОБЛЕНО ТА ВНЕСЕНО: Запорізьким національним технічним університетом
(повне найменування вищого навчального закладу)

РОЗРОБНИКИ ПРОГРАМИ: Федорончак Тетяна Василівна, к.т.н., доцент кафедри ПЗ

Обговорено та рекомендовано до затвердження Вченою радою інституту, факультету, науково- методичною радою

“ _____ ” _____ 2017 року, протокол №__

ВСТУП

Програма вивчення нормативної навчальної дисципліни «Проектний практикум» складена відповідно до освітньо-професійної програми підготовки бакалаврів спеціальності 121 «Інженерія програмного забезпечення».

Предметом вивчення навчальної дисципліни є теоретичні основи та практичні аспекти проектування програмних продуктів.

Міждисциплінарні зв'язки: Курс базується на знаннях, одержаних при вивченні дисциплін «Основи програмної інженерії», «Основи програмування», «Об'єктно-орієнтоване програмування», «Групова динаміка та комунікації». Отриманні знання будуть використовуватися та доповнюватися при подальшому вивченні дисциплін «Архітектура та проектування програмного забезпечення», «Якість програмного забезпечення та тестування», «Аналіз вимог до програмного забезпечення».

Програма навчальної дисципліни складається з таких змістових модулів:

1. Шаблони SOLID та GRASP
2. Шаблони проектування GOF
3. Анті-шаблони

1. Мета та завдання навчальної дисципліни

1.1. Метою викладання навчальної дисципліни “Проектний практикум” є надання майбутньому фахівцю чіткого розуміння про методи і засоби створення програмного проекту на всіх етапах життєвого циклу та їх взаємозв'язок, можливості практично засвоїти їх в проектуванні програмного забезпечення.

1.2. Основними завданнями вивчення дисципліни “Проектний практикум” є опанування основних принципів проектування класів в об'єктно-орієнтованому програмуванні та шаблонів проектування.

1.3. У результаті вивчення навчальної дисципліни студент повинен:

знати :

- теоретичні засади використання шаблонів проектування при розробці архітектури програмного продукту;
- принципи проектування SOLID;
- загальні шаблони розподілу обов'язків GRASP;
- шаблони проектування GOF;
- переваги та недоліки використання шаблонів проектування;
- анті-шаблони в програмуванні;

вміти :

- застосовувати та створювати компоненти багаторазового використання;
- проектувати архітектуру системи з використанням шаблонів проектування.

На вивчення навчальної дисципліни відводиться 90 години/ 3 кредити ECTS.

2. Інформаційний обсяг навчальної дисципліни

Змістовий модуль 1. Шаблони SOLID та GRASP

Основні принципи проектування класів SOLID.

Принципи єдиної відповідальності, відкритості/закритості, підстановки Барбара Лісков, розділення інтерфейсів, інверсії залежностей.

Загальні шаблони розподілу обов'язків GRASP.

Основні шаблони: інформаційний експерт, творець, контролер, низька зв'язаність, високе зачеплення. Додаткові шаблони: чиста вигадка, посередник, поліморфізм, приховування реалізації.

Змістовий модуль 2. Шаблони проектування GOF

Основи шаблонів і їх структури.

Огляд шаблонів проектування. Структура шаблону з чотирьох частин: імені, формулювання завдання, описів рішення і результатів. Переваги використання шаблонів проектування при проектуванні структури ПЗ.

Класифікація шаблонів. Використання шаблонів в проектуванні структури ПЗ.

Класифікація шаблонів проектування: Структурні шаблони, шаблони поведінки та породжуючі шаблони. Сумісне використання шаблонів.

Змістовий модуль 2. Анті-шаблони.

Анті-шаблони в управлінні розробкою програмного забезпечення. Анті-шаблони в об'єктно-орієнтованому програмуванні. Методологічні анті-шаблони.

3. Рекомендована література

1. Guide to the Software Engineering Body of Knowledge (SWEBOK) [Електронний ресурс]. – 2004. – Режим доступу: <http://www.computer.org/portal/web/swebok/html/contents>, <http://swebok.sorlik.ru>.
2. Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE 2004) [Електронний ресурс]. – 2004. – Режим доступу: <http://sites.computer.org/ccse/>
3. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, 2016. – 366 с.
4. Мартин Р. К., Быстрая разработка программ. Принципы, примеры, практика / Р. К. Мартин, Дж. В. Ньюкирк, Р. С. Косс. – М.: Вильямс, 2004. – 752 с.
5. Ларман К. Применение UML и шаблонов проектирования / К. Ларман. – 2-е изд. – М.: Вильямс, 2002. – 624 с.
6. Дубина О. Обзор паттернов проектирования [Електронний ресурс] / О. Дубина. – 2005. – Режим доступу: <http://citforum.ru/SE/project/pattern/#1>
7. Зандстра Мэтт. PHP: объекты, шаблоны и методики программирования / Мэтт Зандстра. – 2-е изд. – М.: Вильямс, 2009. – 480 с. – <http://www.williamspublishing.com/PDF/978-5-8459-1586-3/part.pdf>
8. Фаулер Мартин. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / Мартин Фаулер. – 3-е издание. – СПб: Символ-Плюс, 2011. – 192 с.
9. Соммервилл Иан. Инженерия программного обеспечения / Иан Соммервилл. – 6-е изд. – М.: Вильямс, 2002. – 624 с.
10. Брауде Э. Технология разработки программного обеспечения / Эрик Дж. Брауде. – СПб: Питер, 2004. – 655 с.
11. Марк Гранд Шаблоны проектирования в JAVA. Каталог популярных шаблонов проектирования, проиллюстрированных при помощи UML. – М.: «Новое знание», 2004. – 560 с.

4. Форма підсумкового контролю успішності навчання залік

5. Засоби діагностики успішності навчання Діагностика знань студентів здійснюється з допомогою: завдань для лабораторних робіт; усних опитувань; письмових тестових та контрольних робіт.