



Едуард МАНЗЮК
Олександр БАРМАК
Павло РАДЮК
Марина МОЛЧАНОВА
Олександр ПАСІЧНИК
Руслан БАГРІЙ

ТЕХНОЛОГІЇ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ

ТЕОРЕТИЧНИЙ КУРС ТА
ЛАБОРАТОРНИЙ ПРАКТИКУМ:
НАВЧАЛЬНИЙ ПОСІБНИК



Манзюк Е. А., Бармак О. В., Радюк П. М.,
Молчанова М. О., Пасічник О. А., Багрій Р. О.

**ТЕХНОЛОГІЇ СТВОРЕННЯ
ПРОГРАМНИХ ПРОДУКТІВ.
ТЕОРЕТИЧНИЙ КУРС
ТА ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Навчальний посібник

Хмельницький 2024

УДК 004.4
Т38

*Рекомендовано Вченою радою
Хмельницького національного університету
як навчальний посібник для студентів ЗВО
(лист 041/1693 від 04.06.2024)*

Авторський колектив:

- Манзюк Е. А.**, д-р техн. наук, проф. (вступ, теми 1, 7);
Бармак О. В., д-р техн. наук, проф. (теми 2, 8);
Радюк П. М., д-р філософії, ст. викл. (тема 3);
Молчанова М. О., асистент (тема 4);
Пасічник О. А., канд. техн. наук, доц. (тема 5);
Багрій Р. О., канд. техн. наук, доц. (тема 6)

Рецензенти:

- Остапов С. Е.** – д-р фіз.-мат. наук, проф., зав. кафедри програмного забезпечення комп'ютерних систем, Чернівецький національний університет імені Юрія Федьковича;
Бичков О. С. – д-р техн. наук, проф., лауреат Державної премії в галузі науки і техніки України, зав. кафедри програмних систем і технологій, Київський національний університет імені Тараса Шевченка;
Говорущенко Т. О. – д-р техн. наук, проф., зав. кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет.

АВТОРСЬКИЙ ОРИГІНАЛ-МАКЕТ

Т38 Технології створення програмних продуктів. Теоретичний курс та лабораторний практикум : навч. посіб. / Е. А. Манзюк, О. В. Бармак, П. М. Радюк та ін. Хмельницький : ХНУ, 2024. 306 с.

ISBN 978-617-7600-48-9

Подано теоретичний матеріал та лабораторний практикум з дисципліни «Технології створення програмних продуктів». Усі теми включають питання для самостійного контролю. Виконання лабораторних робіт та самостійних завдань, спрямоване на практичне засвоєння отриманих теоретичних знань.

Для студентів першого рівня вищої освіти, які навчаються за спеціальністю 122 «Комп'ютерні науки».

УДК 004.4

ISBN 978-617-7600-48-9

© Авторський колектив, 2024
© ХНУ, оригінал-макет, 2024

ЗМІСТ

ПЕРЕДМОВА	6
РОЗДІЛ І. ТЕОРЕТИЧНИЙ КУРС	9
ТЕМА 1. Шаблони проєктування GoF. Породжуючі шаблони	9
1.1. Вступ в шаблони проєктування	9
1.2. Абстрактна фабрика (Abstract Factory)	11
1.3. Будівельник (Builder)	17
1.4. Метод фабрики (Factory Method)	23
1.5. Прототип (Prototype)	29
1.6. Одинак (Singleton)	33
Запитання для самоконтролю	37
Завдання на лабораторний практикум	37
ТЕМА 2. Шаблони проєктування GoF. Поведінкові шаблони	38
2.1. Ланцюг відповідальності (Chain of Responsibility)	38
2.2. Команда (Command)	42
2.3. Інтерпретатор (Interpreter)	47
2.4. Ітератор (Iterator)	60
2.5. Посередник (Mediator)	63
2.6. Зберігач (Memento)	68
2.7. Спостерігач (Observer)	72
2.8. Стан (State)	76
2.9. Стратегія (Strategy)	81
2.10. Метод шаблону (Template Method)	83
2.11. Відвідувач (Visitor)	85
Запитання для самоконтролю	91
Завдання на лабораторний практикум	91
ТЕМА 3. Шаблони проєктування GoF. Структурні шаблони	92
3.1. Адаптер (Adapter)	92
3.2. Міст (Bridge)	94
3.3. Декоратор (Decorator)	98
3.4. Компоновщик (Composite)	102
3.5. Фасад (Facade)	107
3.6. Заступник (Proxy)	110
3.7. Пристосованець (Flyweight)	114
Запитання для самоконтролю	119
Завдання на лабораторний практикум	120
ТЕМА 4. Розширення шаблонів GoF	121
4.1. Віртуальний заступник (Virtual Proxy)	121
4.2. Керування кешем (Cache Management)	126

4.3. Пул об'єктів (Object Pool)	135
4.4. Динамічне компонування (Dynamic Linkage)	141
Запитання для самоконтролю	147
Практичні завдання для самостійного опрацювання	148
ТЕМА 5. Шаблони для конкуруючих операцій.....	149
5.1. Однопотокowe виконання (Single Threathed Execution).....	149
5.2. Об'єкт блокування (Lock Object).....	156
5.3. Охоронюване припинення (Guarder Suspension)	160
5.4. Скасування (Balking).....	166
5.5. Планувальник (Scheduler).....	169
5.6. Блокування читання/запису (Read/Write Lock).....	175
5.7. Виробник-Споживач (Producer-Consumer).....	179
5.8. Двофазне завершення (Two-Phase Termination).....	182
5.9. Подвійна буферизація (Double Buffering)	185
5.10. Асинхронна обробка (Asynchronous Processing).....	189
5.11. Майбутнє (Future).....	197
Запитання для самоконтролю	202
Практичні завдання для самостійного опрацювання	202
ТЕМА 6. Проектування на основі обов'язків (Responsibility-Driven Design, RDD).....	203
6.1. Принципи GRASP	204
6.2. Принцип «Низька зв'язність» (Low Coupling)	205
6.3. Принцип «Високе зачеплення» (High Cohesion).....	209
6.4. Принцип «Інформаційний експерт» (Information Expert)	212
6.5. Принцип «Творець (Creator)»	218
6.6. Принцип «Чиста синтетика» (Pure Fabrication)	219
6.7. Принцип «Перенаправлення» (Indirection).....	222
6.8. Принцип «Поліморфізм» (Polymorphism)	224
6.9. Принцип «Захищені зміни» (Protected Variations).....	226
6.10. Принцип «Контролер» (Controller)	230
Запитання для самоконтролю	236
Практичні завдання для самостійного опрацювання	237
ТЕМА 7. Архітектура корпоративних застосувань. Шаблони логіки сутності	238
7.1. Модель предметної області (Domain Model).....	238
7.2. Шар служб (Service Layer).....	241
7.3. Модуль таблиці (Table Module).....	243
7.4. Сценарій транзакцій (Transaction Script)	246
Запитання для самоконтролю	248
Практичні завдання для самостійного опрацювання	248

ТЕМА 8. Архітектура корпоративних застосувань. Шаблони архітектури джерел даних	249
8.1. Активний запис (Active Record).....	249
8.2. Перетворювач даних (Data Mapper).....	251
8.3. Шлюз запису даних (Row Data Gateway)	255
8.4. Шлюз таблиці даних (Table Data Gateway)	258
Запитання для самоконтролю	260
Практичні завдання для самостійного опрацювання	261
ПЕРЕЛІК ЛІТЕРАТУРИ ДО ТЕОРЕТИЧНОГО КУРСУ	262

РОЗДІЛ II. ЛАБОРАТОРНИЙ ПРАКТИКУМ263

ЛАБОРАТОРНА РОБОТА №1 Застосування породжуючих шаблонів Abstract Factory і Builder	264
ЛАБОРАТОРНА РОБОТА №2 Застосування породжуючих шаблонів Factory Method, Prototype, Singleton.....	267
ЛАБОРАТОРНА РОБОТА №3 Застосування поведінкових шаблонів Chain of Responsibility, Command, Interpreter.....	271
ЛАБОРАТОРНА РОБОТА №4 Застосування поведінкових шаблонів Iterator, Mediator, Memento	276
ЛАБОРАТОРНА РОБОТА №5 Застосування поведінкових шаблонів Observer, State, Strategy	280
ЛАБОРАТОРНА РОБОТА №6 Застосування поведінкових шаблонів Visitor, Template Method, структурного шаблону Adapter	284
ЛАБОРАТОРНА РОБОТА №7 Застосування структурних шаблонів Bridge, Composite, Decorator.....	287
ЛАБОРАТОРНА РОБОТА №8 Застосування структурних шаблонів Facade, Flyweight, Proxy.....	291
САМОСТІЙНА РОБОТА №1 Практичне використання шаблонів проектування для розробки програмних систем	294
САМОСТІЙНА РОБОТА №2 Реалізація багатозадачності з використанням конкуруючих операцій.....	296
САМОСТІЙНА РОБОТА №3 Проектування об'єктно-орієнтованих систем за принципами GRASP.....	298
САМОСТІЙНА РОБОТА №4 Розробка корпоративного застосунку з використанням шаблонів логіки сутності	301
САМОСТІЙНА РОБОТА №5 Розробка корпоративного застосунку з використанням шаблонів архітектури джерел даних	303
ПЕРЕЛІК ЛІТЕРАТУРИ ДО ЛАБОРАТОРНОГО ПРАКТИКУМУ	305

ПЕРЕДМОВА

На поточному етапі розвитку інформаційних технологій та активному впровадженні систем із штучним інтелектом у різні галузі людської діяльності зростає потреба в розробці складних та спеціалізованих програмних систем. Це також призводить до нових та високих вимог щодо підготовки студентів, які обирають спеціальність 122 "Комп'ютерні науки".

Сучасний етап розвитку інформаційних технологій та інтенсивне впровадження шаблонів GoF (Gang of Four) у проєктування програмного забезпечення призводить до необхідності розробки комплексних систем з використанням визнаних патернів. Зокрема, шаблони GoF дозволяють ефективно створювати та управляти об'єктами, що визначається високою ступенем абстракції та гнучкістю архітектури. Принципи GRASP (General Responsibility Assignment Software Patterns) також виявляються критичними у сучасній парадигмі програмного проєктування. Наприклад, практичне застосування дозволяє розмішувати відповідальність за обробку інформації в об'єктах, які володіють нею; визначити, як об'єкти контролюють взаємодію між іншими об'єктами, що підвищує чіткість та ефективність системи. Шаблони для конкуруючих операцій набувають особливої важливості в умовах постійних змін та невизначеності умов. Вони дозволяють реалізувати гнучкість системи до змін у середовищі. Застосування архітектури корпоративних застосувань, в саме шаблонів логіки сутності дозволяють визначати структуру та взаємодію ключових елементів системи. Це сприяє ефективному управлінню даними та транзакціями. Шаблони архітектури джерел даних вирішують завдання управління даними в корпоративних застосуваннях, допомагаючи подолати їхню складність та вирішити проблеми невизначеності. Загальна компетентність у використанні цих шаблонів та принципів надає можливість розробки програмних продуктів, які відповідають сучасним вимогам та легко інтегруються з новими технологіями в різних галузях людської діяльності.

Мета дисципліни "Технології створення програмних продуктів" полягає у формуванні в студентів навичок та розуміння щодо застосування різних шаблонів проєктування у розробці програмного забезпечення. Дисципліна спрямована на розвиток навичок оптимального використання шаблонів для поліпшення архітектури, ефективного управління кодом та створення високоякісних програмних продуктів. Дисципліна спрямована на навчання студентів використовувати шаблони проєктування для створення програм, які легко розширюються та адаптуються до змін. Основний акцент робиться на забезпеченні повторного використання коду, оптимального використання ресурсів та створенні програмних продуктів, які відповідають вимогам сучасних стандартів та ринкових потреб. У підсумку, мета полягає в

формуванні компетентностей, необхідних для успішної роботи у сфері розробки програмних продуктів, з використанням передових методів та практик.

Освоєння дисципліни у повному обсязі сприяє розширенню навичок та досягнень у навчанні, набуття компетентностей та програмних результатів навчання, зокрема таких як: здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління; розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук; здатність аналізувати вимоги до програмних продуктів, виявляти проблеми та визначати ефективні рішення; здатність визначати та застосовувати відповідні шаблони проєктування для поліпшення архітектури та структури програм; навички написання ефективного та оптимізованого коду, що сприяє підвищенню продуктивності та якості програм; здатність працювати у команді, використовуючи принципи колективного програмування; вміння створювати гнучкі та масштабовані архітектури програм, які легко адаптуються до змін вимог та ринкових умов; знання та вміння використовувати сучасні мови програмування, фреймворки та інструменти розробки; здатність планувати та приймати стратегічні рішення у процесі розробки програмних продуктів.

Таким чином, розвинуті компетентності, які формуються в процесі навчання, дозволяють студентам успішно впроваджувати передові технології, ефективно розробляти високоякісне програмне забезпечення. Студенти отримують необхідні навички та знання для роботи у галузі розробки програмних продуктів. Ці компетентності дозволяють студентам ефективно впроваджувати сучасні технології, розробляти високоякісне програмне забезпечення та успішно працювати у галузі розробки програмних продуктів.

Структура та зміст курсу спрямовані на студентів, що вивчають спеціальність 122 "Комп'ютерні науки", відповідають актуальним глобальним тенденціям у розробці програмного забезпечення. Курс широко охоплює ключові концепції та методології, які вже визнані стандартами в галузі комп'ютерних наук. Курс враховує внесок висококваліфікованих фахівців, які активно працюють у галузі проєктування та розробки програмних систем на міжнародному рівні: «Design Patterns and Principles in Java» (Ville Korhonen) [2], «Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition» (Mark Grand) [4], «Applied Java Patterns» (Stephen Stelting, Olav Maassen) [9], «Mastering Concurrency Programming with Java 9 - Second Edition: Fast, reactive and parallel application development»

(Javier Fernandez Gonzalez) [10], «Patterns of Enterprise Application Architecture» (Martin Fowler) [12], «Principle Based Enterprise Architecture: A Systematic Approach to Enterprise Architecture and Governance» (Ian Koenig) [13]. Використання викладачами актуальних розробок та підходів, які визнані експертами світового рівня, забезпечує студентам доступ до передових знань та найновіших тенденцій у світі розробки програмного забезпечення.

Навчальний посібник структурно складається з передмови та двох основних розділів зі списками літератури до кожного.

Розділ «Теоретичний курс» містить основні теми: шаблони проектування GoF; породжуючі шаблони, поведінкові шаблони, структурні шаблони; розширення шаблонів GoF; шаблони для конкуруючих операцій; проектування на основі обов'язків; архітектура корпоративних застосувань; шаблони логіки сутності, шаблони архітектури джерел даних. В кінці кожної теми наводяться запитання для самоконтролю, посилання на відповідну лабораторну роботу та завдання для самостійної роботи.

Розділ "Лабораторний практикум" містить завдання щодо виконання лабораторних робіт і самостійних завдань, спрямованих на закріплення теоретичних знань. Метою цих завдань є розвиток практичних навичок у студентів, зокрема вміння проектувати, розробляти програмне забезпечення на базі ефективних практик.

Лабораторні роботи орієнтовані на створення програмного застосунку для управління особистою інформацією, що включає функції адресної книги, особистого планувальника та диспетчера зустрічей і контактів. У цьому контексті детально розглядаються шаблони GoF, де надається програмний код для наочного відображення застосування цих шаблонів. Лабораторні роботи, які рекомендуються в методичних вказівках, спрямовані на розвиток ключових компетенцій у студентів, зокрема, формування алгоритмічних навичок та практичних навичок реалізації методів для вирішення конкретних завдань у сфері розробки програмних систем. Основна мета полягає в тому, щоб студенти отримали необхідні інструменти та знання для успішної роботи над проектами та вирішення прикладних задач у галузі програмування. Програми, розроблені у рамках лабораторних робіт, становлять інструмент для вирішення конкретних завдань у сфері управління особистою інформацією, використовуючи передові шаблони проектування GoF. Лабораторні роботи спрямовані на розвиток ключових навичок та компетенцій у студентів, стимулюючи їхні алгоритмічні та практичні здібності. Основна мета – забезпечити студентам необхідні знання та інструменти для успішної роботи над проектами у сфері програмного забезпечення та ефективного вирішення завдань програмування.

Навчальне видання

*Манзюк Едуард Андрійович,
Бармак Олександр Володимирович,
Радюк Павло Михайлович
та ін.*

**Технології створення
програмних продуктів.
Теоретичний курс
та лабораторний практикум**

Навчальний посібник

Підписано до друку 17.09.2024. Формат 60 × 84 1/16. Папір офсетний.
Друк цифровий. Гарн. Таймс. Ум. друк. арк. 17,6
Тираж 100 прим. Зам. № 2409

Віддруковано: ФОП Мельник А.А.
м. Хмельницький, вул. Чорновола 37, тел. (0382) 74-32-22
Свідоцтво Державного комітету інформаційної політики, телебачення
та радіомовлення України про внесення суб'єкта видавничої справи
до державного реєстру видавців, виготівників та розповсюджувачів видавничої продукції
ДК №1942 від 15.09.2008



МАНЗЮК

Едуард Андрійович



БАРМАК

Олександр Володимирович



РАДЮК

Павло Михайлович



МОЛЧАНОВА

Марина Олексіївна



ПАСІЧНИК

Олександр Анатолійович



БАГРІЙ

Руслан Олександрович

На поточному етапі розвитку інформаційних технологій та активному впровадженні систем із штучним інтелектом у різні галузі людської діяльності зростає потреба в розробці складних та спеціалізованих програмних систем. Запропонований увазі читача навчальний посібник ставить за мету розвиток навичок оптимального використання шаблонів для поліпшення архітектури програмних систем, ефективного управління кодом та створення високоякісних програмних продуктів, які відповідають сучасним вимогам та легко інтегруються з новими технологіями в різних галузях людської діяльності.