

продуктів (проектних планів, документації, програмного коду, тестів, документації користувача тощо) [3].

Однак при формуванні загальних підходів до створення алгоритмічного та програмного забезпечення виникають труднощі, пов'язані з необхідністю досягнення їхньої наочності, структурованості, спостережливості та керованості. Ці проблеми посилюються тим, що при створенні різних систем управління зазвичай використовуються різні технології алгоритмізації та програмування. Спектр таких технологій широкий: від технологій на основі алгоритмічних мов високого рівня для промислових комп'ютерів технологій та на основі спеціалізованих мов для програмованих логічних контролерів (наприклад, функціональних та сходових схем).

Етап програмування включає вибір мови програмування та власне формалізоване (на основі пропонованих методів) ізоморфне програмування за побудованими алгоритмічними моделями. На основі запропонованого підходу розроблена програмна оболонка, що дозволяє проводити моделювання у термінах автоматів. Таким чином, використання запропонованої технології, яка отримала назву SWITCH технологія [4], дозволяє з єдиних позицій проводити алгоритмізацію та програмування при створенні систем логічного управління та знизити кількість помилок в алгоритмах та програмах. Ця технологія може бути названа також STATE-технологія або більше точно AUTOMATON-технологія, а відповідна область програмування – автоматне програмування.

Реалізація того чи іншого завдання може бути проведена безліччю способів, і алгоритмізація, будучи невід'ємною частиною процесу розробки ПЗ, дозволяє розробнику найбільш грамотно і повно оцінити ситуацію і вибрати найкраще рішення. При цьому коректно побудований алгоритм дозволяє розробникам наочно побачити структуру майбутнього проекту та проаналізувати можливі складнощі та перешкоди.

#### Список літератури

1. Дж. Ріхтер. *Windows для професіоналів Створення ефективних WIN32-програм з урахуванням специфіки 64-розрядної версії Windows.*
2. Кьюу Дж., Джеаніні М. *Об'єктно-орієнтоване програмування.*
3. Шалито А.А. *Програмна реалізація керуючих автоматів. Суднобудівна промисловість. Сер. "Автоматика та телемеханіка". 1991. Вип.13.*

**УДК 004.4**

*Рудкевич Б. М., студент 2 курсу спеціальності 122 «Комп'ютерні науки»*

*Потапова Н. А., к.е.н., доцент, доцент кафедри інформаційних технологій*

## АЛГОРИТМІЗАЦІЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

На сьогоднішній день важко уявити життя людини без автоматизованих технологій. Адже щодня кожен з нас користується пристроями різного роду призначення. Зручністю використання та побудовою принципів роботи і займається розробка програмного забезпечення та її алгоритмізація. Розглянемо більш детально.

Розробка програмного забезпечення та побудова безпосередньо самих алгоритмів є головною складовою будь-якого готового програмного продукту. Алгоритмізація дає змогу описати процеси мовою математичних символів для одержання безпосередньо самого алгоритму. В свою чергу розробка програмного забезпечення ставить собі за мету, створення та подальшу підтримку працездатності, якості та надійності програмного забезпечення. Для цього застосовують різноманітні технології з таких сфер, як інформатика, проектування, управління цифровими даними і дизайн у всіх його проявах. [1]

Розробка будь-якої програми, будь-то невелика процедура обробки даних отриманих від користувача або ж багаторівневий комплексний програмний продукт, складається з певних етапів, коректна реалізація яких, є обов'язковою умовою для отримання хорошого результату в подальшому.

Тепер мабуть варто перейти до розгляду самої структури побудови ПЗ. формування вимог проекту:

1. Постановка мети. На першому етапі, фахівець даної предметної області повинен визначити чітко мету, її зміст і загальний підхід до її вирішення

2. Проектування. На другому етапі формується модель вирішення даної задачі, проте на відміну від першого, тут її формуванням займаються спеціалісти ПЗ, адже на даному кроці потрібно розбити кінцевий проєкт на підпрограми, визначити загальний дизайн та спосіб зберігання інформації

3. Реалізація. Лише на цьому кроці можна приступати безпосередньо до реалізації, тобто написання коду за яким і буде виконана вся програма

4. Тестування. Під час тестування фахівці, також програмної області, повинні перевірити на справність кожен частину готового програмного продукту та дати свій експертний висновок, чи готовий він до видання

5. Впровадження. На передостанньому етапі розробники передають продукт замовнику, якщо такий є.

6. Експлуатація та супровід. Останній етап розробки програмного продукту це експлуатація та супровід, який включає в себе консультацію користувачів по роботі з продуктом та усунення помилок в процесі експлуатації.

Звісно для кожного проєкту можуть і мають бути свої відмінності, проте в загальному вони мають наступний вигляд:

У відповідності до вище зазначених пунктів в основному у розробці програмного забезпечення беруть участь: користувач, замовник, дизайнер, керівник проєкту, аналітик, тестувальник та постачальник.

Після безпосередньої розробки продукту слід проаналізувати його якість. Тобто те, наскільки чітко він дозволяє задовольнити потреби користувача. Лише після того, як програма стане корисною користувачу, тобто матиме попит, можна говорити про повне завершення розробки та алгоритмізації.

Отож, розкриваючи суть питання можна відмітити, що розробка та алгоритмізація програмного продукту це багатофункціональний процес, у якому беруть участь спеціалісти з різних галузей, кожен з яких виконує свою частину роботи для подальшого успіху проєкту та можливості його покращення в майбутньому.

### Список літератури

1. Дегтярьова Л.М., Гроза П.М., Сомов С.В. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». Полтава: ПолтНТУ, 2017. 218 с.
2. Крєневич А.П. Алгоритми і структури даних. Підручник. К.: ВПЦ "Київський Університет", 2021. 200 с.
3. Трофименко О.Г., Козін О.Б., Задерейко О.В., Плачінда О.Є. Веб-технології та веб-дизайн: навч. посібник. Одеса: Фенікс, 2019. 284 с.

**УДК 004.43**

*Солодун Т. Р., студентка 3 курсу  
Ніколюк П.К., д. ф.-м.н., професор,  
професор кафедри інформаційних  
технологій*

## ОСОБЛИВОСТІ ВИКОРИСТАННЯ MIDDLEWARE В МОВІ ПРОГРАМУВАННЯ JAVA

*Донецький національний університет імені Василя Стуса, м. Вінниця*

*Анотація.* У даному дослідженні подана інформація про поняття проміжного програмного забезпечення. Методологічною основою роботи є розгляд транзакцій у мові програмування Java. Специфіка досліджуваної теми передбачає розгляд даного поняття на прикладі використання у реалізації робочих проєктів.

*Ключові слова:* middleware, java, JDBC.

Для початку варто ознайомитись з основними поняттями для кращого розуміння даної теми. Отже, Middleware – це програмний рівень, розташований між додатками та операційними системами. Проміжне програмне забезпечення зазвичай використовується в розподілених системах, чим спрощує розробку програмного забезпечення [1].