

УДК 004.01

*Поліщук А.М, бакалавр, КН-2А,  
Ніколюк П.К, д-р фізико-  
математичних наук, професор*

## **ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ ТА ОСНОВНІ ЕТАПИ ЇЇ РОЗВИТКУ**

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

Програмування — це процес розробки, написання, тестування та підтримки комп'ютерних програм, яке поєднує в собі елементи інженерії, фундаментальної науки та мистецтва. Загалом кажучи, воно охоплює розробку, алгоритми та аналіз майбутніх вимог користувачів програмного забезпечення. Серед основних особистих якостей, необхідних для програмування, є пам'ять, логічне мислення, концентрація та здатність виконувати велику кількість дрібних завдань.

Техніки програмування — це набір технічних інструкцій, які включають порядок виконання технічних операцій, перелічення умови, за яких виконується та чи інша операція та опис самої операції, де дані та результати визначені для кожної операції. Існує різниця між методами, які використовуються на певному етапі розробки, і методами, які охоплюють кілька етапів або весь процес розробки. Перший базується на обмежено застосовних методах, які дозволяють вирішувати конкретні проблеми. Другий — базовий метод або підхід, який визначає набір методів, що використовуються на різних етапах розробки.

Програми для перших комп'ютерів створювалися в машинному коді або на мові асемблера. Пошук помилок у програмі можна порівняти з тортурами Тантала. Програмісти - це як "вища каста" серед звичайних людей, тому що вони єдині, хто може спілкуватися з комп'ютерним обладнанням. Цей етап програмування називається «спонтанним програмуванням». Створення мов програмування високого рівня, таких як FORTRAN і ALGOL, значно спрощує обчислювальне програмування. Це дає можливість збільшити складність програми. Створено цілу бібліотеку сервісів і обчислювальних процедур, які можна використовувати в різних програмних системах. Дані програми зберігаються в глобальній зоні. У 1960-х вибухнула криза програмування. Проблема більшості проектів програмного забезпечення полягає у великій кількості помилок, а пошук і виправлення цих помилок вимагає багато часу розробки. Багато з них так і не були завершені. Причиною цього є відсутність продуманого технічного чи програмного підходу. Це призвело до створення робочої групи з методів програмування в Міжнародній федерації з обробки інформації. До неї входять багато відомих програмістів, таких як К. Хоар та Е. Дейкстра. Їх спільні зусилля привели до створення нової техніки програмування – структурного програмування.

Другий етап – це набір рекомендованих технічних прийомів, що охоплюють виконання всіх етапів розробки програмного забезпечення. Основою

є декомпозиція складної системи, метою якої є подальша реалізація у вигляді окремих невеликих підпрограм. Структурні підходи вимагають представлення завдань у вигляді ієрархії підзадач у їх найпростішій структурі. Проектування «зверху вниз», орієнтоване на реалізацію загальної ідеї, що забезпечує розвиток інтерфейсу підпрограми. Підтримка принципів структурованого програмування базується на так званих процедурних мовах програмування. Вони включають базові «структурні» оператори передачі керування, підтримуване рутинне вкладення та локалізацію. Одночасно зі структурним програмуванням з'явилася велика кількість мов, заснованих на інших концепціях. Деякі мови були забуті повністю, інші розвинули свої ідеї далі і зберегли їх у наступній версії. Зі збільшенням складності розроблюваного програмного забезпечення виникла необхідність у розробці структуризації даних. Як наслідок, у мові можуть бути визначені власні типи даних. Зростає бажання розмежувати доступ до глобальних програмних даних аби зменшити помилки. В результаті почала зароджуватися і розвиватися техніка модульного програмування, яка розділяє групи підпрограм, що використовують одні й ті ж глобальні дані, на окремо скомпільовані модулі. Модульне програмування значно спрощує розробку програмного забезпечення кількома програмістами. Програмісти самостійно розробляють власні модулі, а взаємодія між модулями забезпечується спеціально визначеними міжмодульними інтерфейсами. Модулі можуть бути використані в подальшому без необхідності внесення змін до інших розробок, що підвищує продуктивність праці програмістів. Вузким місцем модульного програмування є те, що помилки інтерфейсу при виклику підпрограм виявляються лише під час виконання програми. Об'єктно-орієнтоване програмування-це метод створення складного програмного забезпечення, заснований на представленні програми у вигляді сукупності об'єктів, де кожен об'єкт є екземпляром певного типу, а класи успадковують властивості, утворюючи ієрархію. Об'єктна структура програм вперше була прийнята в Simula-мові імітаційного моделювання складних систем. Природний спосіб представлення програм на мовах моделювання був прийнятий у нових версіях мов програмування загального призначення, таких як Pascal, C++, Modula та Java. Основна перевага об'єктно-орієнтованого програмування перед модульним полягає в тому, що воно "природніше" декомponує програмне забезпечення і значно полегшує його розробку. Стрімкий розвиток методів програмування на основі об'єктного підходу дозволив вирішити багато проблем. При використанні цих мов програмування зберігається залежність програмних модулів від структур і форматів даних, а також від адрес експортованих полів і методів. Ця залежність є об'єктивною, оскільки модулі повинні взаємодіяти між собою з посиланням на ресурси один одного. Хоча взаємозв'язок між модулями може бути порушений, існує можливість стандартизації їх взаємозв'язків.

Останній етап-компонентний підхід та CASE-технології. Компонентний підхід-це метод побудови програмного забезпечення з окремих компонентів. Об'єкти-компоненти збираються в динамічно викликані бібліотеки або виконувані файли, розповсюджуються в двійковому вигляді і можуть використовуватися на будь-якій мові програмування, що підтримує відповідну

технологію. Компонентний підхід покладено в основу технологій, розроблених на базі COM і CORBA. Ці технології використовують схожі принципи і відрізняються лише функціональністю їх реалізації. Технологія COM є розвитком технології OLE I, яка використовувалася для створення складних документів, яка є взаємодією між усіма типами програм, включаючи бібліотеки, додатки та операційні системи. Об'єкт завжди функціонує у складі сервера.

Існує три типи серверів: внутрішні сервери-реалізуються динамічними бібліотеками, які підключаються до клієнтських додатків і працюють в одному з ними адресному просторі, локальні сервери-створюються іншим процесом, запущеним на тому ж комп'ютері, що і клієнт та віддалений сервер, що створюється процесом, запущеним на іншому комп'ютері.

Особливістю розвитку сучасних технологій програмування є створення та реалізація технологій автоматизації розробки та супроводу програмного забезпечення, відомих як CASE-технології. Сьогодні існують CASE-технології, які підтримують як структурний, так і об'єктно-орієнтований підходи.

#### Список літератури.

1. Зубенко В. В. *Програмування: навчальний посібник (гриф МОН України) / В. В. Зубенко, Л. Л. Омельчук. — К. : ВПЦ «Київський університет», 2011. — 623 с.*
2. Нікітченко — Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010. — 121 с.
3. У. Дал, Е. Дейкстра, К. Хоор. *Структурне програмування. — М.: Мир, 1973. — 247 с.*

#### УДК 004.4

*Проценко А. С., студентка 1 курсу  
спеціальності 122 «Комп'ютерні науки»  
Гончар В. М., асистент  
кафедри інформаційних технологій*

### **СТВОРЕННЯ МАКРОСІВ ДЛЯ АВТОМАТИЗАЦІЇ ОДНОТИПНИХ ЗАВДАНЬ В MICROSOFT EXCEL**

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

При створенні електронних таблиць часто необхідно виконувати однотипні завдання по кілька разів. Задля спрощення виконуваної роботи та зменшення навантаження на користувача можна застосувати макроси.

Макроси – набір дій та певних інструкцій, що об'єднані в команду для машинової реалізації завдання. Вони дозволяють заощаджувати час виконання операцій. Макрос можна застосовувати стільки разів, скільки це необхідно для виконання завдання, створивши його лише один раз. Характер застосованих операцій може бути різнобічним. До прикладу, форматування клітинок, рядків, стовпців, введення чисел чи тексту, натискання певних клітинок або команд на стрічці, імпортування даних із зовнішніх джерел.

У середовищі Microsoft Excel макроси можна записувати за допомогою спеціального засобу запису – Macro Recorder, включеної в табличний процесор,