**Лабораторна робота №09 Процес розробки програмного забезпечення**

**На період** карантину в дистанційній формі навчання на надані запитання потрібно надати письмові відповіді, надіславши їх на електронну адресу викладача. Файл надавати з іменем у форматі

**PI<Номер групи><Номер лекції / практичної / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, P –практична, R – лабораторна]<Прізвищеанглійською>**. Наприклад, **PI3104L**buts.doc. Відповіді на запитання повинні бути не довгими і змістовними. Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності відповідей-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк виконання цієї роботи ІПЗ-31, ІПЗ-32 18.05.2020**

**Мета:** Навчиться оформляти результати проектування з погляду на вимоги до програмного забезпечення.

**Завдання:** узагальнити результати своєї роботи над проектом в одному документі, доповнивши його новими розділами.

**Методичні вказівки**

1. Продовжуємо працювати з постановкою задачі з минулих ЛР та ПР по створенню застосунку для викладачів та студентів гіпотетичного навчального закладу або теми власної курсової роботи.

Очікуваний підсумковий результат щодо проведеної роботи (що повинно бути зроблено) надано в таблиці:

| ЛР/ПР | Результат |
| --- | --- |
| ПР№2 | – Назва проекту, його мета, результати і вхідні дані, основні функції, які реалізовує ПЗ у вигляді структурної схеми –Модель ЖЦ ПЗ та обґрунтування вибору – Схема реалізації моделі ЖЦ ПЗ – Склад бізнес-процесів основних та допоміжних, вказавши перелік допоміжних процесів на кожному етапі ЖЦ ПЗ |
| ПР№3 | - |
| ЛР№02 | –Діаграма прецедентів (варіантів / випадків використання) –Таблиця з описом вимог (табл.1 ) |
| ПР№5 | Діаграма класів |
| ЛР№03 | Діаграма послідовностей для кожного раніш створеного прецеденту |
| ПР№6 | Уточнені та узгоджені діаграми прецедентів, класів та послідовностей |
| ЛР№4 | Діаграма кооперації, уточнені діаграми прецедентів та послідовностей |
| ПР№7 | –Діаграма діяльності –Доповнена таблиця з описом вимог (табл.1 ) |
| ЛР№5 | - |
| ПР№8 | - |
| ЛР№6 | - |
| ПР№9 | - |
| ПР№10 | –Документ „Вимоги користувача”,  –Документ „Функціональні вимоги до програмного забезпечення” |
| ЛР№7 | –Таблиця "Способи перевірки нефункціональних вимог"  –Діаграми прецедентів та кооперації з прив’язкою об’єктів та дій до конкретної вимоги, внесені до документу „Вимоги користувача” –Доопрацьований документ „Функціональні вимоги до програмного забезпечення” |
| ЛР№8 | Доопрацьовані документи „Вимоги користувача”, „Функціональні вимоги до програмного забезпечення” |
| ПР№11 | – Окремий розділ "Основні технічні рішення " – Окремий розділ "Специфікації процесів" – Доопрацьовані документи „Вимоги користувача”, „Функціональні вимоги до програмного забезпечення” |
| ПР№12 |

1. Всі матеріали оформіть одним документом, перевірте узгодженість всіх частин. Матеріали ПР№2 оформлюйте як перший розділ Загальні відомості" та другий "Модель ЖЦ ПЗ" з відповідними підрозділами. Цей документ буде вами в подальшому використовуватися в ході вивчення програмної інженерії та конструювання програмного забезпечення, доповнюватися та удосконалюватися.
2. Надайте письмові відповіді на запитання
3. Що в серед наданих матеріалів (див. п.2) відноситься до поведінкових нотацій, а що до структурних?
4. Які на Вашу думку дії з конструювання Ви вже реалізували в своєму проекті?
5. До роботи повинен надаватися звіт з титульним листом із визначенням «Лабораторна робота № », після цього написати назву системи / застосунку для якого відносяться матеріали.
6. По закінченню лабораторну роботу потрібно здати на перевірку викладачеві, надіславши електронною поштою. Якщо викладач знаходить помилки чи неточності, він може повернути роботу на доопрацювання.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**Проектування ПЗ** - це процес визначення архітектури, набору компонентів, їх інтерфейсів, інших характеристик системи і кінцевого складу програмного продукту.

Область знань «Проектування ПЗ (Software Design)» складається з таких розділів:

- базові концепції проектування ПЗ (Software Design Basic Concepts),

- ключові питання проектування ПЗ (Key Issue in Software Design),

- структура й архітектура ПЗ (Software Structure and Architecture),

аналіз і оцінка якості проектування ПЗ (Software Design Quality Analysis and Evaluation),

- нотації проектування ПЗ (Software Design Notations),

- стратегія і методи проектування ПЗ (Software Design Strategies and Methods).

**Базова концепція проектування ПЗ** - це методологія проектування архітектури за допомогою різних методів (об'єктного, компонентного й ін.), процесів ЖЦ (стандарт ISO/IEC 12207) і техніки - декомпозиція, абстракція, інкапсуляція й ін. На початкових стадіях проектування предметна область декомпозується на окремі об'єкти (при об'єктно-орієнтованому проектуванні) або на компоненти (при компонентному проектуванні). Для подання архітектури програмного забезпечення вибираються відповідні артефакти (нотації, діаграми, блок-схеми і методи).

**Ключові питання проектування** - це декомпозиція програм на функціональні компоненти для незалежного і одночасного їхнього виконання, розподіл компонентів у середовищі функціонування і їх взаємодія між собою, забезпечення якості і живучості системи й ін.

**Проектування архітектури ПЗ** проводиться архітектурним стилем, заснованим на визначенні основних елементів структури - підсистем, компонентів, об'єктів і зв'язків між ними.

**Архітектура проекту** - високорівневе подання структури системи і специфікація її компонентів. Архітектура визначає логіку системи через окремі компоненти системи настільки детально, наскільки це необхідно для написання коду, а також визначає зв'язки між компонентами. Існують і інші види подання структур, засновані на проектуванні зразків, шаблонів, сімейств програм і каркасів програм.

Один з інструментів проектування архітектури - **патери (шаблон).** Це типовий конструктивний елемент ПЗ, що задає взаємодію об'єктів (компонентів) проектованої системи, а також ролі і відповідальності виконавців. Основна мова опису - UML. Патерн може бути **структурним,** що містить у собі структуру типової композиції з об'єктів і класів, об'єктів, зв'язків і ін.; **поведінковим,** що визначає схеми взаємодії класів об'єктів і їх поведінку, задається діаграмами адитивностей, взаємодії, потоків керування й ін.; **погоджувальним,** що відображає типові схеми розподілу ролей екземплярів об'єктів і способи динамічної генерації структур об'єктів і класів.

**Аналіз і оцінка якості проектування ПЗ** - це заходи щодо аналізу сформу­льованих у вимогах атрибутів якості, функцій, структури ПЗ, з перевірки якості ре­зультатів проектування за допомогою метрик (функціональних, структурних і ін.) і методів моделювання і прототипування.

**Нотації проектування** дозволяють представити опис об'єкта (елемента) ПЗ і його структуру, а також поведінку системи. Існує два типи нотацій: структурна, поведінкова, та множина їх різних представлень.

**Структурні нотації -** це структурне, блок-схемне або текстове подання аспектів проектування структури ПЗ з об'єктів, компонентів, їх інтерфейсів і взаємозв'язків. До нотацій відносять формальні мови специфікацій і проектування: ADL (Architecture Description Language), UML (Unified Modeling Language), ERD (Entity-Relation Diagrams), IDL (Interface Description Language) тощо. Нотації містять y собі мовний опис архітектури й інтерфейсу, діаграм класів і об'єктів, діаграм сутність-зв'язок, конфігурації компонентів, схем розгортання, а також структурні діаграми, що задають у наочному вигляді оператори циклу, розгалуження, вибору і послідовності.

**Поведінкові нотації** відбивають динамічний аспект роботи системи та її ком­понентів. Ними можуть бути діаграми потоків даних (Data Flow), діяльності (Activity), кооперації (Colloboration), послідовності (Séquence), таблиці прийняття рішень (Décision Tables), передумови і посту мови (Pre-Post Conditions), формальні мови специфікації (Z, VDM, RAISE) і проектування.

**Стратегія і методи проектування ПЗ.** До стратегій відносять: проектування вгору, вниз, абстрагування, використання каркасів і ін. Методи є функціонально-орієнтовані, структурні, які базуються на структурному аналізі, структурних картах, діаграмах потоків даних й ін.

Вони орієнтовані на ідентифікацію функцій і їх уточнення знизу-вгору, після цього уточнюються діаграми потоків даних і проводиться опис процесів.

В об'єктно-орієнтованому проектуванні ключову роль відіграє спадкування, поліморфізм й інкапсуляція, а також абстрактні структури даних і відображення об'єктів. Підходи, орієнтовані на структури даних, базуються на методі Джексона і використовуються для подання вхідних і вихідних даних структурними діаграмами. Метод UML призначений для опису сценаріїв роботи проекту у наочному діаграмному вигляді. Компонентне проектування ґрунтується на використанні готових компонентів (reuse) з визначеними інтерфейсами і їх інтеграції в конфігурацію, як основи розгортання компонентної системи для її функціонування в операційному середовищі.

Формальні методи опису програм ґрунтуються на специфікаціях, аксіомах, описах деяких попередніх умов, твердженнях і посту мовах, що визначають заключну умову одержання правильного результату програмою. Специфікація функцій і даних, якими ці функції оперують, а також умови і твердження — основа доведення правильності програми.

Конструювання програмного забезпечення (software construction) - детальне створення робочої програмної системи з конструкцій (блоків, операторів, функцій) за допомогою комбінації кодування, верифікації (перевірки), модульного тестування (unit testing), інтеграційного тестування та відлагодження.

З конструюванням програмного забезпечення пов’язані конкретні задачі:

* Перевірка виконання умов, необхідних для успішного конструювання.
* Проектування і написання класів та методів.
* Створення та присвоєння імен змінним та іменованим константам.
* Вибір управляючих структур та організація блоків команд.
* Модульне тестування, інтеграційне тестування та відлагодження власного коду.
* Взаємний огляд коду та низькорівневих програмних структур членами групи.
* Інтеграція програмних компонентів, створених окремо.
* Оптимізація коду, направлена на підвищення його швидкодії і зниження степені використання ресурсів.

Конструювання займає у середньому 30-80% від загального часу, затраченого на роботу і являється центром процесу розробки програмного забезпечення. Це єдиний процес, що в будь-якому випадку виконується при написанні програмного забезпечення.

Область знань «Конструювання ПЗ (Software Construction))) містить у собі такі розділи:

- зниження складності (Reduction in Complexity),

- попередження відхилень від стилю (Anticipation of Diversity),

- структуризація перевірок (Structuring for Validation),

- використання стандартів (Use of External Standards).

**Зниження складності** - це мінімізація, зменшення і локалізація складності конструювання.

**Мінімізація складності** - це обмеження на обробку складних структур і вели­ких обсягів інформації протягом тривалого періоду часу. Вона досягається, зокрема, використанням у процесі конструювання простих елементів, а також рекомендацій стандартів.

**Зменшення складності** в конструюванні ПЗ досягається шляхом створення простого коду, що легко читається і спрощує тестування, підвищує продуктивність і впливає на досягнення інших характеристик і обмежень проекту. Зменшення складності спрощує процеси верифікації і тестування результатів конструювання елементів ПС.

**Локалізація складності —** це спосіб конструювання з застосуванням об'єктно-орієнтованого підходу, що лімітує інтерфейс об'єктів, спрощує їхню взаємодію, перевірку правильності самих об'єктів і зв'язків між ними. Локалізація призначена для внесення змін, пов'язаних з виявленими помилками в коді, або коли джерелом помилок є середовище, у якому виконується код.

**Попередження відхилень від стилю.** Для розв'язання різних задач конструювання застосовуються різні стилі конструювання (лінгвістичний, формальний, візуальний).

**Лінгвістичний стиль** заснований на використанні словесних інструкцій і виразів для подання окремих елементів (конструкцій) програм. Він призначений для конструювання нескладних конструкцій і приводиться до вигляду традиційних функцій і процедур або реалізується методами логічного і функціонального програмування й ін.

**Формальний стиль** використовується для точного й однозначного визначення компонентів системи, мінімальної кількості помилок, що можуть виникнути в зв'язку з неоднозначністю визначень або невдалих узагальнень об'єктів конструювання ПЗ.

**Візуальний стиль** - найбільш універсальний для конструювання прикладного ПЗ. Він дозволяє представляти елемент конструювання у наочному вигляді. Візуальна мова проектування UML надає розробнику набір діаграм для подання статичної і динамічної структур ПЗ. При його застосуванні створюється-текстовий і діаграмний опис конструктивних елементів ПЗ, який виводиться на екран дисплея для перегляду і коригування.

**Структуризація перевірок** припускає, що побудова ПС структурована таким чином, що спрощується пошук помилок, дефектів і різних збоїв у процесі перевірок як на стадії незалежного тестування, так і в процесі експлуатації. Структуризації перевірок сприяють огляд, інспектування, спільний перегляд, модульне тестування із застосуванням автоматизованих засобів тестування й ін.

**Використання зовнішніх стандартів.** Конструювання ПЗ залежить від застосовних зовнішніх стандартів, пов'язаних з мовами програмування, інструментальними засобами й інтерфейсами. При конструюванні має бути визначений достатній набір стандартів для керування і забезпечення координації між визначеними видами діяльності і групами операцій, мінімізації складності, внесення змін, аналізу ризиків тощо.

До таких стандартів відносять: мови програмування (Java, Ада 95, C++ і ін.), інтерфейси мов програмування (МП) і прикладні інтерфейси платформ Windows (COM, DCOM), CORBA і ін. При конструюванні використовують стандарти мов опису даних (XML, SQL і ін.), засобів комунікації (COM, CORBA. і ін.), інтерфейсних мов (POS1X, IDL, APL), UML і ін.

Перелічені вище розділи області знань «Конструювання ПЗ» у ядрі знань SWEBOK об'єднуються в групу «Основи конструювання». Крім того, розглядають­ся групи розділів «Керування конструюванням» та «Практичні міркування». Опи­шемо першу з них детальніше.

**Керування конструюванням** - це керування процесом конструювання ПЗ, планування, оцінка виконання плану і розроблення заходів щодо внесення змін.

**Моделі конструювання** містять у собі набір операцій, послідовність дій і результатів. Види моделей визначаються стандартом ЖЦ, методологіями і практиками. Деякі стандарти ЖЦ за своєю природою орієнтовані на конструювання типу екстремального програмування і раціонального уніфікованого процесу - RUP (Rational Unified Process).

**Планування** - це визначення порядку операцій, термінів і рівня виконання заданих умов у процесі конструкторської діяльності за моделлю ЖЦ, що містить у собі задачі і дії зі створення, перевірки й оцінки показників якості. Виконавці роз­поділяються за процесами і виконують відповідні задачі з реалізації проміжного і кінцевого продукту. Остаточний результат виміряється за обсягом коду, ступенем повторного використання, кількістю помилок і дефектів, а також оцінюються пока­зники якості ПЗ.

**Внесення змін** пов'язане з помилками, виявленими при перевірці і тестуванні, проводиться з метою збереження функціональної цілісності системи. У випадку виявлення помилок на етапі супроводження приймається рішення про внесення змін або заміну коду у цілому.