

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Компоненти програмної інженерії 1.

Вступ до програмної інженерії»

«Специфікування предметної галузі проєкту

засобами мови UML»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірів Марченко Олена Іванівна

Київ 2022

Лабораторна робота 3

Специфікування предметної галузі проєкту засобами мови UML

Мета: дослідження класів та отримання навиків у побудові діаграми класів UML для специфікування предметної галузі, використанні стереотипів UML та структуруванні моделі UML за допомогою пакетів.

Завдання.

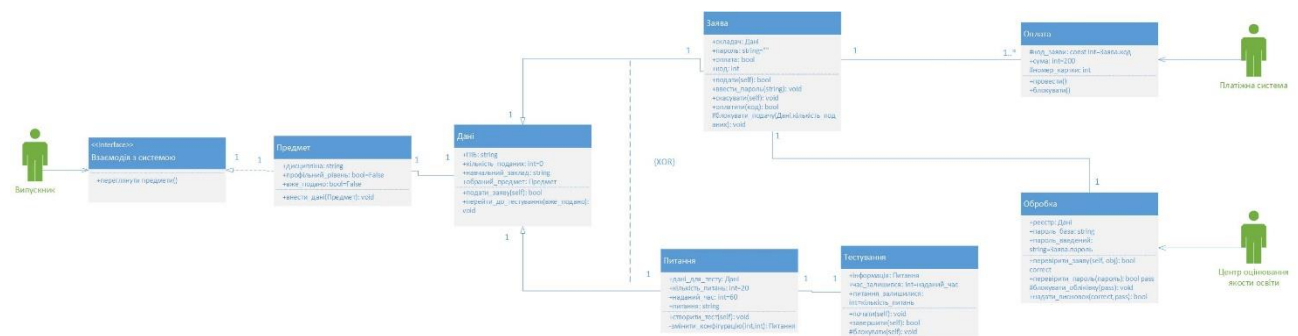
1. Виявити класи, які відносяться до системи, що проєктується (мінімум 5 класів). Намагатися використовувати повний синтаксис. Стереотипи класів використовувати там, де вони потрібні! В інших випадках беремо стандартне зображення класу.
2. Коротко текстом описати призначення та використання кожного класу та атрибутів і операцій.
3. Намагатися використати усі вивчені зв'язки у діаграмі (у всіх темах вони є).

Варіант №18.

18	Інформаційна система ЗНО	дозволяє проводити реєстрацію абітурієнтів на певний вид екзаменів, проводити їх
----	-----------------------------	---

Хід роботи.

1. Визначимо класи для проєктованої системи. Відберемо клас інтерфейсу системи «Взаємодія з системою». Також визначимо клас «Предмет» і «Дані», який матиме класи-нащадки «Заява» і «Питання». «Питання» будуть пов'язані із «Тестуванням». «Тестування» буде тупиковим класом на діаграмі, натомість від «Заяви» матимемо зв'язки – класи «Оплата» і «Обробка». Актор Платіжна система взаємодіатиме з класом «Оплата», з «Обробкою» буде мати справу актор Центр оцінювання якості освіти.
2. Побудуємо діаграму класів.



(Діаграма також буде прикріплена окремо.)

3.
 - 1) Клас інтерфейсу «Взаємодія з системою». Призначений для взаємодії користувача з інформаційною системою ЗНО. Його публічний метод – переглянути предмети(), призначений для перегляду предметів учня, що йому доступні. Клас з'єднаний з класом «Предмет» зв'язком реалізації інтерфейсу.
 - 2) Клас «Предмет». Призначений для зберігання загальної інформації, пов'язаної з дисципліною учня і її станом, а також наступної взаємодії щодо дисципліни. Клас з'єднаний зв'язком з класом «Дані». Атрибути класу: дисципліна – змінна рядкового типу, назва дисципліни, про яку йде мова у даному об'єкті класу; профільний рівень – логічна змінна, показує, чи рівень дисципліни для учня був профільним, і відповідно впливає на рівень іспиту (за замовчуванням False – непрофільний); вже_подано – булева змінна, показує, чи була подана вже заява для даного іспиту, і чи можна відповідно подавати її зараз (за замовчуванням False – не подана).
Методи: внести_дані(Предмет): void – призначений для внесення даних про дисципліну до загальних даних користувача і переходу до класу «Дані».

- 3) Клас «Дані». Призначений для зберігання усіх необхідних даних про користувача, а також на основі цієї інформації вибору необхідної наступної взаємодії. Клас з'єднаний зв'язком з класом «Заява» і «Тестування».

Атрибути класу: ПІБ – рядкова змінна, зберігає прізвище, ім'я, по батькові складача; кількість_поданих – ціла змінна, зберігає кількість вже поданих заяв, за замовчуванням 0; навчальний_заклад – рядковий тип, зберігає заклад, де навчався випускник; обраний_предмет – рядковий тип, зберігає предмет, заяву якого розглядатимемо зараз.

Методи: подати_заяву() – призначений для подання заяви на складання іспиту за даним предметом; перейти_до_тестування() – призначений для переходу до меню тестування за даним предметом.

- 4) Клас «Заява». Призначений для керування заявою, діями, пов'язаними з нею, такі як оплата й перевірка коректності. Клас з'єднаний з класом «Оплата» й «Обробка».

Атрибути класу: складач – змінна-об'єкт класу Дані, призначена для збереження даних, що є про користувача для тих чи інших дій із заявою; оплата – булева змінна, показує, чи така заява вже оплачена; код – цілочисельна змінна, зберігає унікальний код заяви; пароль – рядкова змінна, призначена для зберігання внесеного паролю для перевірки, якщо не внесено – за замовчуванням пустий рядок.

Методи: подати() – призначений для подання заяви, передаються необхідні дані, інформація заявника, заповнюються автоматично деякі поля, необхідні для формування заяви і передання її на перевірку; скасувати() – призначений для скасування подання заяви, так само, як і в попередньому методі, передаються необхідні дані і подання на розгляд; оплатити() – призначений для переходу до меню оплати, після якої можна буде пройти тестування; захищений – блокувати_подачу(Дані.кількість_поданих) – призначений для блокування можливості подачі заяв за виконання деяких умов, як-от перевищення кількості уже поданих; ввести_пароль(string) – призначений для введення паролю для перевірки.

- 5) Клас «Оплата». Призначений для деталізації оплати, взаємодії системи з актором Системи оплати, передання коштів задля розблокування доступу до тестування. Клас взаємодіє зв'язком з актором Системи оплати.

Атрибути: захищені – код_заяви – унікальний номер заяви, необхідний для проведення оплати, константа; номер_картки – необхідно для доступу до коштів на рахунку платника; сума – кількість коштів, які користувач планує переказати на оплату цієї заяви (за замовчуванням 200 – 200 грн – повна сума). Оскільки сплачувати заяву можна

частинами, то маємо відношення один до багатьох, де на одну заяву може припадати більше однієї оплати;

Методи: провести() – якщо дані введені коректно, платіжна система знімає гроші, перераховує їх на потрібний рахунок, оплата вважається здійсненою; блокувати() – якщо дані тричі введено некоректно, можливість оплати блокується.

- 6) Клас «Обробка». Призначений для перевірки коректності даних, звірення їх з даними реєстрів та надання висновку щодо подання заяви; Клас взаємодіє зв'язком з актором Центру оцінювання якості освіти.

Атрибути: реєстр – дані типу Дані, що надійшли з реєстру для перевірки заяви; пароль_база – рядкові дані для перевірки паролю з введеним; пароль_введений – рядкові дані, що дорівнюють введеному паролю у класі Дані.

Методи: перевірити_заяву() – звіряє дані складача з даними реєстру; перевірити_пароль() – звіряє дані облікового запису, якщо пароль невірний тричі, викликає метод блокувати_обліківку() – блокує обліковий запис користувача і системі як підозрілий; надати_висновок() – відповідно до не\коректності даних, надає висновок про не\успішну реєстрацію заяви як для системи, так і для користувача шляхом надсилення листа.

- 7) Клас «Питання». Призначений для зберігання інформації про формат тесту, кількість питань, час тощо, а також переходу до класу Тестування.

Атрибути: дані_для_тесту – передає інформацію з класу Дані; кількість_питань – відображає кількість питань, яка може варіюватися залежно від рівня профілю (за замовчуванням – 20, стандартна к-сть); наданий_час – аналогічно до кількості питань – може варіюватися (за замовчуванням 60 – 60 хв – стандартна тривалість). питання – зберігає посилання на питання, що виберуться кількістю випадковим чином.

Методи: створити_тест() – формує тест, вибирає питання, кількість, час на основі дані_для_тесту, готує систему до початку тесту.

Приватний – змінити_конфігурацію() – змінює час або кількість питань у зв'язку із помилкою або втручанням ззовні.

- 8) Клас «Тестування». Призначений для проходження тестування з даного предмету.

Атрибути: інформація – передає інформацію з класу Питання;

час_залишився – час, що залишився для тестування, змінюється за допомогою таймеру в методі почати(); питання_залишилися – маркує уже вибрані відповіді;

Методи: почати() – відкриває тестування, починає відлік заданого часу, дає можливість зберігати відповіді; завершити() – закінчує

тестування через кінець часу або натисканням відповідної кнопки до кінця, передає відповіді на сервер, закриває можливість їх змінення, відбувається обробка; блокувати – системою блокується можливість тестування з ануляцією відповідей через несанкціоноване втручання чи інші причини.

Клас «Питання» і «Заява» поєднані XOR-асоціацією, адже в даний момент часу можна використовувати лише дії з одного з класів.

Висновок.

Отже, у цій роботі я дослідив класи та отримав навички у побудові діаграми класів UML для специфікування предметної галузі, використанні стереотипів UML та структуруванні моделі UML за допомогою пакетів. У результаті лабораторної роботи було проведено аналіз, визначено класи для проєктованої системи, побудовано діаграму класів на основі проведеного попереднього аналізу, а також проведено опис використання класів. Використовуючи засоби специфікування й програмний засіб MS Visio для побудови діаграми, отримуємо коректний результат.