

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №3 з дисципліни
«Компоненти програмної інженерії 1.
Вступ до програмної інженерії»

«Специфікування предметної галузі
проекту засобами мови UML»
Варіант 3

Виконав студент ІП-15, Борисик Владислав Тарасович
(шифр, прізвище, ім'я, по батькові)

Перевірила Марченко Олена Іванівна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №3
Специфікування предметної галузі проекту засобами мови uml
Варіант 3
Завдання
Завдання

1. Виявити класи, які відносяться до системи що проектується (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. (для здачі без захисту. Альтернативний варіант - усно при захисті розказати).

3. Намагатися використати усі вивчені зв'язки діаграми (у всіх темах вони є).

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.

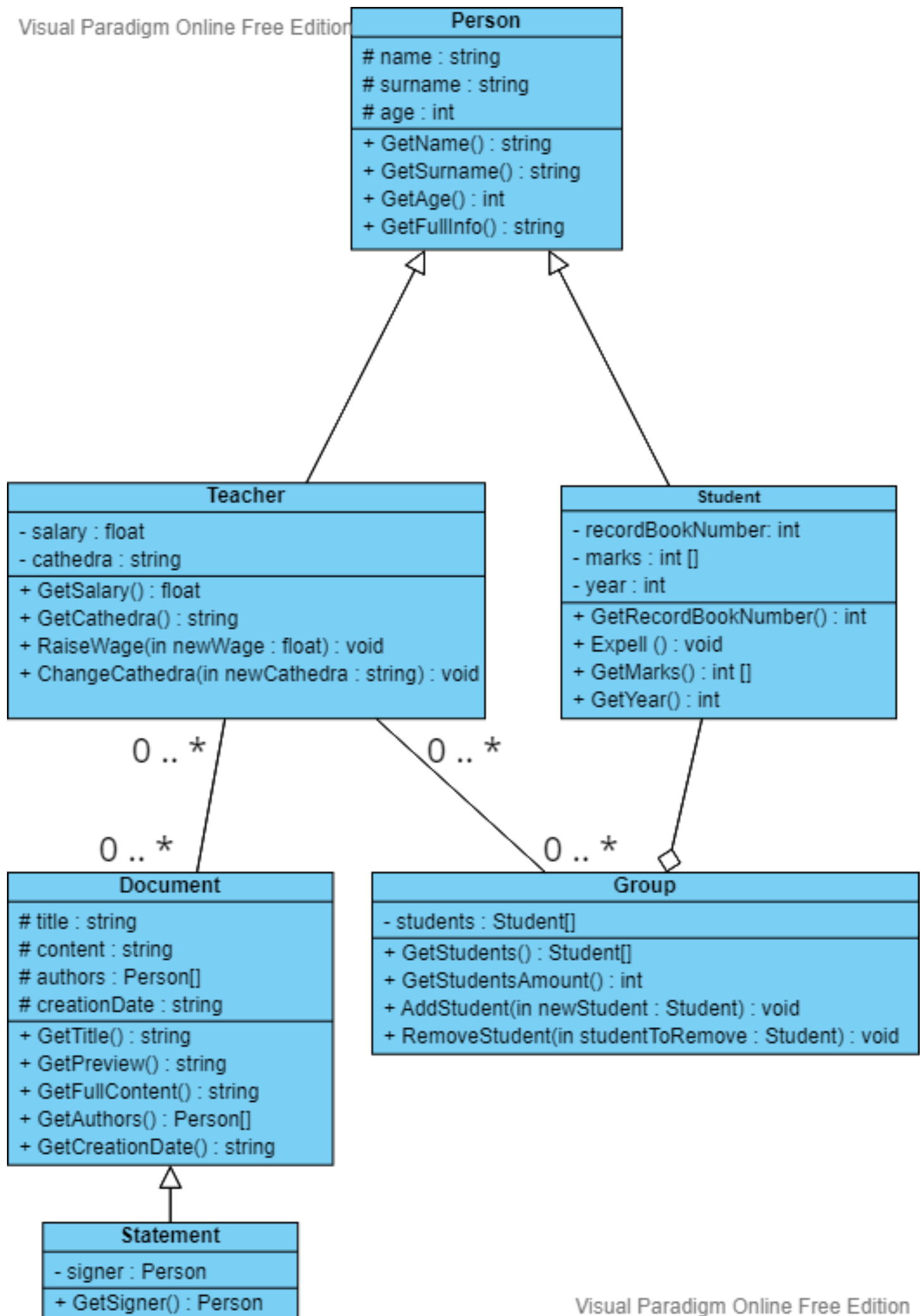
В цій лабораторній роботі я буду проектувати інформаційну систему деканату.

Класифікація класів, які відносяться до системи що проектується

Для проектування інформаційної системи деканату я виявив 6 класів.

1. Особа (Person)
2. Студент (Student)
3. Викладач (Teacher)
4. Група (Group)
5. Документ (Document)
6. Відомість (Statement)

Діаграма класів



Опис призначення та використання класів, їх атрибутів і операцій

1) Клас Person (Особа)

Має 3 атрибути:

1. name (ім'я) - призначений для зберігання імені особи. Має специфікатор доступу protected. Тип даних - string.
2. surname (прізвище) - призначений для зберігання прізвища особи. Має специфікатор доступу protected. Тип даних - string.
3. age (вік) - призначений для зберігання віку особи. Має специфікатор доступу protected. Тип даних - string.

Має 4 методи:

1. GetName() - призначений для отримання імені особи. Має специфікатор доступу public. Не приймає аргументів. Повертає ім'я особи з типом даних string.
2. GetSurname() - призначений для отримання прізвища особи. Має специфікатор доступу public. Не приймає аргументів. Повертає прізвище особи з типом даних string.
3. GetAge() - призначений для отримання віку особи. Має специфікатор доступу public. Не приймає аргументів. Повертає вік особи з типом даних int.
4. GetFullInfo() - призначений для отримання повної інформації про особу. Має специфікатор доступу public. Не приймає аргументів. Повертає повну інформацію про особу з типом даних string.

2) Клас Teacher (Викладач) успадкований з класу Person

Має 2 атрибути:

1. salary (заробітня плата) - призначений для зберігання заробітньої плати викладача. Має специфікатор доступу private. Тип даних - float.
2. cathedra (кафедра) - призначений для зберігання назви кафедри, на якій працює викладач. Має специфікатор доступу private. Тип даних - string.

Має 4 методи:

1. GetSalary() - призначений для отримання заробітньої плати викладача. Має специфікатор доступу public. Не приймає аргументів. Повертає заробітню плату викладача з типом даних float.
2. GetCathedra() - призначений для отримання назви кафедри, на якій працює викладач. Має специфікатор доступу public. Не приймає аргументів. Повертає назву кафедри з типом даних string.

3. RaiseWage(in newWage : float) - призначений для підвищення заробітної плати викладача. Має специфікатор доступу public. Приймає 1 аргумент: newWage - нова заробітна плата викладача з типом даних float. Нічого не повертає.
4. ChangeCathedral(in newCathedral : string) - призначений для зміни назви кафедри, на якій працює викладач. Приймає 1 аргумент: newCathedral - нова назва кафедри з типом даних string. Нічого не повертає.

3) Клас Student (Студент) успадкований з класу Person

Має 3 атрибути:

1. recordBookNumber (номер залікової книжки) - призначений для зберігання номеру залікової книжки. Має специфікатор доступу private. Тип даних - int.
2. marks (оцінки) - призначений для зберігання оцінок студента. Має специфікатор доступу private. Тип даних - масив з цілочисельними значеннями (int[]).
3. year (рік, курс навчання) - призначений для зберігання курсу, на якому навчається студент. Має специфікатор доступу private. Тип даних - int.

Має 4 методи:

1. GetRecordBookNumber() - призначений для отримання з номеру залікової книжки. Має специфікатор доступу public. Не приймає аргументів. Повертає номер залікової книжки з типом даних int.
2. Expell () - призначений для відрахування студента. Має специфікатор доступу public. Не приймає аргументів. Нічого не повертає.
3. GetMarks() - призначений для отримання оцінок студента. Має специфікатор доступу public. Не приймає аргументів. Повертає оцінки студента з типом даних масив з цілочисельними значеннями (int[]).
4. GetYear() - призначений для отримання курсу, на якому навчається студент. Не приймає аргументів. Повертає курс на якому навчається студент з типом даних int.

4) Клас Document (Документ)

Має 4 атрибути:

1. title (заголовок) - призначений для зберігання заголовку документу. Має специфікатор доступу protected. Тип даних - string.
2. content (зміст) - призначений для зберігання змісту документу. Має специфікатор доступу protected. Тип даних - string.
3. authors (автори) - призначений для зберігання авторів документу. Має специфікатор доступу protected. Тип даних - масив з класів Person (Person[]).

4. `creationDate` (дата створення документу) - призначений для зберігання дати створення документу. Має специфікатор доступу `protected`. Тип даних - `string`.

Має 5 методів:

1. `GetTitle()` - призначений для отримання заголовку документу. Має специфікатор доступу `public`. Не приймає аргументів. Повертає заголовок документу з типом даних `string`.
2. `GetPreview()` - призначений для отримання короткого змісту документа. Має специфікатор доступу `public`. Не приймає аргументів. Повертає короткий зміст документа з типом даних `string`.
3. `GetFullContent()` - призначений для отримання повного змісту документа. Має специфікатор доступу `public`. Не приймає аргументів. Повертає повний зміст документа з типом даних `string`.
4. `GetAuthors()` - призначений для отримання авторів документа. Не приймає аргументів. Повертає масив з класів `Person` (`Person[]`).
5. `GetCreationDate()` - призначений для отримання дати створення документу. Не приймає аргументів. Повертає дату створення документу з типом даних `string`.

5) Клас `Statement` (Відомість) успадкований з класу `Document`

Має 1 атрибут:

1. `signer` (підписувач) - призначений для зберігання особи, яка підписала відомість. Має специфікатор доступу `private`. Тип даних - `Person`.

Має 1 метод:

1. `GetSigner()` - призначений для отримання особи, яка підписала відомість. Має специфікатор доступу `public`. Не приймає аргументів. Повертає особу, яка підписала документ з типом даних `Person`.

6) Клас `Group` (Група)

Має 1 атрибут:

1. `students` (студенти) - призначений для зберігання студентів. Має специфікатор доступу `private`. Тип даних - масив з класів `Student` (`Student []`).

Має 4 методи:

1. `GetStudents()` - призначений для отримання студентів. Має специфікатор доступу `public`. Не приймає аргументів. Повертає масив з класів `Student` (`Student []`).

2. `GetStudentsAmount()` - призначений для отримання кількості студентів в групі. Має специфікатор доступу `public`. Не приймає аргументів. Повертає кількість студентів в групі з типом даних `int`.
3. `AddStudent(in newStudent : Student)` - призначений для додавання студента в групу. Має специфікатор доступу `public`. Приймає 1 аргумент: `newStudent` - студент, якого потрібно додати з типом даних `Student`. Нічого не повертає.
4. `RemoveStudent(in studentToRemove : Student)` - призначений для видалення студента з групи. Приймає 1 аргумент: `studentToRemove` - студент, якого потрібно видалити з типом даних `Student`. Нічого не повертає.

Висновок

Під час виконання цієї лабораторної роботи 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.