

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА
Географічний факультет
Кафедра геодезії, картографії та управління територіями**

МОВА ПРОГРАМУВАННЯ PYTHON, ЯК ІНСТРУМЕНТ ПІДВИЩЕННЯ ЦИФРОВОЇ КОМПЕТЕНТНОСТІ СПЕЦІАЛІСТА В ГАЛУЗІ ГЕОДЕЗІЯ

**Кваліфікаційна робота
Рівень вищої освіти - другий (магістерський)**

Виконав :
студент II курсу, групи 618
спеціальності 193 «Геодезія та землеустрій»
ОПП «Геодезія»
ШИЙЧУК Тарас Тарасович
(прізвище, ім'я та по-батькові)
Керівник: к.геогр.н., доц. Дарчук К. В.
(науковий ступінь, вчене звання, прізвище та ініціали)

До захисту допущено:
протокол засідання кафедри № ____
від «__» «_____» 2024 р.
Зав. кафедрою _____ доц. Дарчук К.В.

Чернівці – 2024

АНОТАЦІЯ

Шийчук Т.Т.

Мова програмування PYTHON, як інструмент підвищення цифрової компетентності спеціаліста в галузі «геодезія»

Кваліфікаційна робота зі спеціальності
193 «Геодезія та землеустрій» ОПП «Геодезія»

Ключові слова: Python, геоінформаційні системи, цифрова компетентність, QGIS, автоматизація.

У кваліфікаційній роботі розглянуто використання мови програмування Python як інструменту для формування цифрових компетентностей у геодезії. Досліджено перспективи Python у підготовці спеціалістів, зокрема для автоматизації робочих процесів. Розроблено плагін для програмного забезпечення QGIS, який автоматизує процес передачі результатів лабораторних робіт через Telegram-бот. Робота містить практичні рекомендації щодо інтеграції Python у навчальний процес для підвищення ефективності підготовки майбутніх фахівців.

ABSTRACT

Shyichuk T. T.

Python Programming Language as a Tool for Enhancing Digital Competence of Specialists in Geodesy

Qualification work on the specialty 193 "Geodesy and land management" EP
"Geodesy"

Keywords: Python, geoinformation systems, digital competence, QGIS, automation.

The qualification work considers the use of the Python programming language as a tool for the formation of digital competencies in geodesy. The prospects of Python in the training of specialists are investigated, in particular for the automation of work processes. A plug-in for the QGIS software has been developed, which automates the process of transmitting laboratory results via a Telegram bot. The work contains practical recommendations for integrating Python into the educational process to increase the efficiency of training future specialists.

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів наукових досліджень інших авторів мають посилання на відповідне джерело.

Тарас ШИЙЧУК

ЗМІСТ

ВСТУП.....	3
Розділ 1. ГЕОДЕЗИЧНА ОСВІТА УКРАЇНИ.....	6
1.1 Історія, суть, призначення та проблеми.....	6
1.2. Цифрова компетентність у геодезії та освітньому процесі.....	10
Розділ 2. АНАЛІЗ ПЕРСПЕКТИВ, МОЖЛИВОСТЕЙ І ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ МОВОЮ PYTHON.....	15
2.1 Мова програмування Python та історія її створення.....	15
2.2 Можливості застосування, перспективи та способи вивчення.....	19
2.3 Python як інструмент для геодезиста.....	25
Розділ 3. ВИКОРИСТАННЯ PYTHON В СТВОРЕННІ ПЛАГІНУ ДЛЯ ПРОГРАМНОГО ПРОДУКТУ QGIS.....	28
3.1 Мета, ціль та опис функціональності розробки.....	28
3.2 Вибір архітектури та технологічного стеку плагіну.....	30
3.3 Встановлення плагіну та ілюстрація його можливостей.....	31
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТОК А. СЕРТИФІКАТИ ПРО ЗАКІНЧЕННЯ КУРСІВ ІЗ МОВИ ПРОГРАМУВАННЯ PYTHON.....	48
ДОДАТОК Б. КОД ПРОГРАМНОГО ПРОДУКТУ.....	52

ВСТУП

Актуальність. Запровадження карантинних обмежень і дистанційного (онлайн) навчання в освітні процеси значно ускладнило його проведення. Це стало особливо помітним у навчальній діяльності через труднощі з адаптацією практичних занять до цифрового формату, їх планування та створення у вигляді електронних уроків та занять [1]. До того ж, в Україні швидко зростає потреба у працівниках та фахівцях, які мають базові знання, уміння та навички в цифрових та інформаційно-комунікаційних технологіях, а знання та вміння застосовувати й розуміти основні принципи цих технологій набуває ключових вимог до майбутніх фахівців будь-якої галузі [2], адже нині такі вміння стають необхідним інструментом у діяльності багатьох спеціалістів та фахівців.

До студентів ВНЗ (вищих навчальних закладів) за спеціальністю «Геодезія» висуваються нові вимоги — їхня підготовка має відповідати тенденціям розвитку цифрового суспільства, основою якого є діджиталізація, що забезпечується належною матеріальною базою та фахівцями. Цифровізація розглядається як ключовий механізм прогресу на шляху до побудови інформаційного суспільства завдяки її здатності впливати на результативність, продуктивність, вартість, якість економічної, особистісної та суспільної діяльності [2, 3]. Усі аспекти життя сучасної людини потребують оновлення змісту стандартів і навчальних програм, зокрема для спеціальностей інженерно-педагогічного напрямку.

Тому впровадження інформаційно-комунікаційних технологій у навчально-освітній процес є важливим завданням, яке може забезпечити високий рівень компетентностей фахівців. Найперспективнішим способом підвищення цифрової компетентності є оволодіння мовами програмування. Цій темі й присвячено дане дослідження.

Мета роботи — пошук та аналіз сучасних інструментів для розвитку цифрових компетентностей.

Завдання роботи:

- 1) висвітлити призначення, суть, та зміст освітньої програми «Геодезія»;
- 2) провести дослідження та аналіз інформації стосовно можливостей використання мови програмування Python;
- 3) дослідити можливості застосування зазначеної мови програмування як інструмента для розвитку цифрових компетентностей фахівців у сфері «Геодезія»;
- 4) запрограмувати власний програмний продукт – плагін для програмного застосунку QGIS – як приклад практичного застосування Python у процесі навчання.

Об’єкт дослідження – процес розвитку цифрових компетентностей.

Предмет дослідження – формування цифрових компетентностей за допомогою мови програмування Python.

Методи дослідження: аналіз літературних джерел, освітніх концепцій і рекомендацій; синтез отриманих знань; педагогічний експеримент.

Компетентність і апробація. У ході виконання роботи автор пройшов навчання за програмами вивчення мови програмування Python на платформах Coursera та Cisco, а також у компаніях Datawiz, SoftServe, Uployal та Monobank, за підсумками кожного з яких було отримано відповідні сертифікати (див. Додаток А).

Практичне значення результатів дослідження полягає у демонстрації можливостей застосування Python в робочих та/або навчальних процесах.

Структура роботи: робота обсягом в 61 сторінку містить вступ і три розділи. У першому розділі розглядаються суть, призначення та проблеми сучасної освіти, а також значення цифрових компетентностей для майбутніх фахівців у галузі геодезії. У другому розділі описується історія створення мови програмування Python, її можливості та перспективи використання як інструмента для підвищення цифрової грамотності. Третій розділ присвячений опису процесу розробки програмного продукту, використаних інструментів, аналізу його функціональності та потенційних способів реалізації.

Також у роботі наведено висновки та список використаних джерел з 42 бібліографічних найменувань, 24 рисунків та 2 додатки.

Розділ 1. ГЕОДЕЗИЧНА ОСВІТА УКРАЇНИ

1.1 Історія, суть, призначення та проблеми

Геодезична освіта в Україні націлена на підготовку спеціалістів, здатних здійснювати точні вимірювання поверхні Землі, визначати її форму та розміри, а також створювати плани і карти для вирішення різноманітних наукових і прикладних завдань. Це охоплює вивчення методів геодезичних вимірювань, обробку геопросторових даних та застосування сучасних геоінформаційних систем (ГІС). Основними завданнями геодезичної освіти є:

- Підготовка кваліфікованих фахівців — забезпечення ринку праці спеціалістами, здатними виконувати геодезичні роботи у будівництві, землеустрої, картографії та суміжних галузях.
- Розвиток наукових досліджень — підтримка наукових досліджень у сфері геодезії, геоінформатики та пов'язаних дисциплін.
- Забезпечення державних потреб - підтримка державних програм, пов'язаних із землеустроєм, кадастром, містобудуванням та обороною.

Геодезична освіта в Україні має давні історичні корені, що сягають кількох століть. Перші відомості про геодезичні знання на українських землях датуються часами Київської Русі, коли для потреб землеволодіння та будівництва застосовувалися прості методи вимірювання.

У XVIII столітті, з розвитком науки та техніки, геодезія почала формуватися як окрема дисципліна. Важливим кроком стало відкриття у 1834 році Київського університету, де географія, включно з геодезією, викладалася як самостійний навчальний предмет. Згодом, у 1882 році, географічні кафедри було відкрито у Львівському університеті, а в 1889 році — у Харківському університеті [4].

У радянський період геодезична освіта зазнала значного розвитку. Було створено спеціалізовані навчальні заклади та кафедри, що готували фахівців для потреб народного господарства. Зокрема, у 1930 році в Києві було створено Інститут інженерів геодезії, аерофотозйомки та картографії, який згодом став головним центром підготовки геодезистів в Україні.

Після здобуття незалежності у 1991 році геодезична освіта в Україні зазнала істотних змін. Відбувся перехід до багаторівневої системи підготовки спеціалістів, що охоплює бакалаврські та магістерські програми. Університети почали впроваджувати сучасні методи навчання, інтегруючи новітні технології, зокрема геоінформаційні системи (ГІС) та супутникові технології.

Нині геодезична освіта в Україні представлена в багатьох вищих навчальних закладах, серед яких Київський національний університет імені Тараса Шевченка (КНУ), Львівський національний університет імені Івана Франка, Національний університет "Львівська політехніка" та інші. Ці заклади пропонують освітні програми, що відповідають сучасним вимогам ринку праці та сприяють розвитку геодезичної науки в Україні.



Рис. 1.1. Львівська політехніка кафедра геодезії

Сучасний розвиток освітньої системи відбувається в надзвичайно складних умовах, які характеризуються зростанням вимог до підготовки кваліфікованих кадрів, а також загостренням економічних ризиків, демографічних проблем, соціокультурних та етичних аспектів [5, 6]..

У таких умовах невід'ємною умовою розвитку освіти стає інноваційний процес, особливістю якого є технологізація. Головне завдання цього процесу полягає в забезпеченні навчального середовища сучасними освітніми ресурсами та підготовці фахівців до їх ефективного застосування. Швидкий розвиток цифрових технологій, що супроводжується постійною розробкою й вдосконаленням програмних засобів із високим навчальним потенціалом, вимагає оновлення змісту освіти. У таких умовах важливо, щоб майбутній фахівець міг формувати свій індивідуальний освітній потенціал для розвитку цифрової грамотності [7–9].

Останнім часом викладачі прагнуть трансформувати навчально-виховні технології в інтерактивні форми, засновані на взаємодії між викладачем і студентом. Цінність такого підходу полягає у розвитку в учнів актуальних сьогодні навичок, таких як критичне мислення, здатність до рефлексії, міжособистісна взаємодія, робота в команді та групі, осмислена продуктивна діяльність, а також усвідомлення критеріїв оцінювання власних досягнень тощо.

З огляду на подальший розвиток технологій, майбутнім фахівцям дедалі більше потрібні будуть міждисциплінарні навички, знання та вміння, які дозволяють інтерпретувати, аналізувати та застосовувати інформацію в різноманітних ситуаціях професійної діяльності, виконуючи складні виробничі завдання. У зв'язку з цим у процесі підготовки кваліфікованих фахівців усе ширше застосовуються інформаційно-комунікаційні (діджитал)технології:

- новітні засоби комунікацій,
- «хмарні» та віртуальні освітні середовища
- анімації,
- відеоконференції,
- Інтернет,
- інтернет-бібліотеки,
- онлайн курси,
- засоби комп'ютерного тестування й автоматизованого програмування

новітньо-технологічних процесів.,

- віртуальні та онлайн мультимедійні музеї, клуби тощо.



Рис. 1.2. Цифрові технології в освіті

Завдяки цим технологіям відбувається перехід до програмної підтримки гейміфікації в навчанні, а також створення віртуальних симуляцій (VR) реального виробничого середовища для забезпечення професійного навчання через цифрові інтернет-ресурси [8]. Очікується, що в майбутньому будь-який фахівець зможе «конструювати» себе з елементів професійно-освітнього конструктора, доступного на ринку праці. Це означатиме, що кожен матиме свій цифровий портрет, що відображає набуті компетентності, на основі яких роботодавці зможуть обирати працівників і вкладати ресурси в їхнє подальше професійне вдосконалення. Використання сучасних програмних засобів навчання також може суттєво покращити якість освіти, адже комп'ютеризація, цифровізація та технологізація значно розширюють інтелектуальні можливості майбутніх спеціалістів [7,12].

1.2. Цифрова компетентність у геодезії та освітньому процесі

У сучасній геодезичній освіті України цифрова компетентність є основним компонентом підготовки фахівців. Володіння сучасними геоінформаційними системами (ГІС) та навички програмування суттєво підвищують ефективність роботи геодезистів. Цифрова компетентність включає здатність до ефективного використання цифрових технологій для збору, обробки та аналізу геопросторових даних. В Україні розвиток цифрових навичок підтримується на державному рівні: Кабінет Міністрів України затвердив Концепцію розвитку цифрових компетентностей до 2025 року, що сприяє підвищенню цифрової грамотності населення [8].

Аналіз перспектив розвитку геодезичної освіти в Україні підкреслює актуальність її модернізації та вдосконалення. У процесі підготовки геодезистів використовуються такі програмні продукти, як QGIS — потужна відкрита геоінформаційна система, широко застосовувана для обробки та аналізу просторових даних. Вона підтримує різні формати даних і надає розширені можливості для візуалізації й аналізу. В Україні також є ініціативи зі створення інструментів для просторового планування в QGIS, що сприяє розвитку цифрових навичок фахівців [8].

Опанування мови програмування Python є важливим для автоматизації завдань і розширення функціональності ГІС. QGIS підтримує інтеграцію з Python, що дозволяє створювати скрипти для автоматизації процесів, розробляти плагіни й налаштовувати програму під специфічні потреби. Для початку роботи з PyQGIS рекомендується ознайомитися з офіційною документацією [9] і пройти відповідні навчальні курси. Комбінація знань у сфері ГІС, програмування на Python та загальної цифрової грамотності є необхідною для сучасного геодезиста в Україні.

Питання інформатизації та цифровізації в останні роки привертають значну увагу багатьох сучасних педагогів, практиків та науковців. Зокрема, дослідження проблем використання комп'ютерних технологій у навчанні є темою праць багатьох українських та зарубіжних вчених, серед яких С.

Яшанова, Р. Вільямс, Б. Гершунський, В. Глушков, Г. Громов, К. Маклін, Є. Машбиц, В. Гриценко, С. Пейперт, Б. Сендов, І. Зязюн, О. Агапов, Б. Хантер, Г. Клейман та інші. Окремі аспекти цифровізації та інформатизації освіти, зокрема професійної підготовки викладача, висвітлені в роботах С. Хоменко, Р. Гуревича, К. Тверезовської, В. Ковальчука, Н. Брюханової та інших.

Важливим аспектом підвищення цифрової грамотності суспільства є цифровізація освіти, яка визначається як «процес, спрямований на задоволення освітніх інформаційних потреб учасників освітнього процесу» [7]. Цифровізація освіти передбачає створення системи, орієнтованої на використання передових інформаційно-комунікаційних технологій у навчальному процесі для його інтенсифікації, реалізації ідей розвивального й адаптивного навчання, а також удосконалення форм і методів організації навчання [10].

Для викладачів геодезії цифрова компетентність є ключовою для підготовки студентів до сучасних вимог ринку праці. Застосування цифрових технологій у навчальному процесі сприяє інтерактивності, доступності та актуальності навчальних матеріалів. Геодезисти, які володіють цифровими навичками, можуть ефективно інтегрувати новітні технології у робочі процеси, що підвищує якість роботи у відповідь на професійні виклики.

Цифровізація навчального процесу спирається на сучасні комп'ютерні та телекомунікаційні технології, базуючись на принципах інтерактивного навчання, що дозволяє ефективно керувати поданням навчального матеріалу. Це надає студентам можливість індивідуально налаштовувати навчальний процес: змінювати швидкість подання матеріалу, обмежувати кількість повторень і пристосовувати процес навчання до особистих здібностей та потреб, бачити результати й оцінку навчання.

Однією з головних переваг планування освіти на основі цифрових технологій, на думку авторів [12], є здатність реалізувати принцип діалогу. Цей підхід забезпечує організацію співпраці педагога зі студентами в аудиторії, створення освітніх міркувань, навчальних діалогів і їх комп'ютерного

моделювання, що підвищує ефективність навчання та взаємодію в освітньому процесі.

Огляд численних літературних джерел з питань цифровізації та інформатизації освіти свідчить про постійне оновлення понятійного апарату, що дозволяє узагальнити поняття «цифрової культури», «цифрової грамотності» та об'єднуюче поняття «цифрової компетентності». Зокрема, у сучасній науковій літературі категорія цифрової культури трактується як культурологічний, соціологічний та технологічний феномен [6-15]. Основне смислове навантаження тут припадає на здатність здійснювати дії за допомогою цифрових пристроїв, появу нових специфічних інформаційно-віртуальних форм культури (такі як комп'ютерна музика, NetArt, віртуальні інсталяції тощо), а також нових способів культурної комунікації (web-спілкування) і видів діяльності (наприклад, IT-волонтерство, «Greening IT» та інші).

Цифрова компетентність зазвичай базується на загальноприйнятому визначенні компетентності як інтегрованої здатності особистості, що включає досвід, уміння, навички, цінності та ставлення, які можуть бути цілісно реалізовані на практиці. У цьому контексті цифрова компетентність є узагальнюючим поняттям для попередніх термінів, оскільки охоплює як цифрову грамотність, так і цифрову культуру [17].

Згідно з сучасними дослідженнями світового ринку праці, станом на 2021 рік частка спеціальностей зросла на 27%. Внаслідок швидкої автоматизації виробничих процесів прогнозується зникнення близько 75 мільйонів робочих місць. Для збереження своєї затребуваності та конкурентоздатності на ринку праці понад 54% працівників змушені пройти перенавчання або підвищити кваліфікацію. Високий рівень безробіття серед молоді зумовлений браком у значної їх частини необхідних професійних навичок.

Це вимагає від геодезистів широкого кола компетентностей для успішного виконання професійних функцій. Швидкозмінні процеси, на які фахівець не може впливати, але має бути до них готовий, також відіграють важливу роль. У 2016-2017 роках європейські наукові спільноти і практики розробили

Європейську рамку цифрової компетентності для громадян, яка визначає цифрову компетентність як здатність використовувати новітні технології для розвитку творчості, громадянської активності, соціальної інтеграції та співпраці для досягнення особистих, соціальних або комерційних цілей. Документ описує цифрову компетентність як поєднання цифрової та інформаційної грамотності, що включає співпрацю і комунікацію, створення цифрового контенту (зокрема програмування), безпеку в мережі й вирішення конфліктів [18]. Така компетентність забезпечує впевнене й критичне застосування ІКТ для створення, пошуку, обміну та обробки інформації в роботі, публічному просторі й приватному спілкуванні.

Сучасні наукові публікації дедалі більше акцентують увагу на поняттях «цифрова компетентність» і «цифрова грамотність». Зокрема, в дослідженні [16] цифрова грамотність описується через наявність у людини таких навичок, як:

- взаємодія із комп'ютерами та/або будь-яким іншими пристроями (hardware skills), за допомогою яких можна користуватися інформаційними мережами або створювати цифрові артефакти;
- взаємодія з ПЗ (програмне забезпечення - software skills), яка забезпечує можливість роботи з контентом;
- практичне застосування цифрових технологій (meta skills), зокрема, технологій для конструювання та розробки цифрових онлайн або офлайн середовищ.

Аналізуючи поняття «цифрова компетентність», можна дійти висновку, що воно охоплює сукупність знань, умінь та навичок для отримання, обробки, передачі та застосування інформації в різних сферах діяльності, що забезпечує ефективне виконання професійних функцій. Формування та вдосконалення такої компетентності у майбутніх геодезистів тісно пов'язане з прагненням до навчання впродовж життя, підвищенням професійної майстерності, що є важливим фактором якісної професійної освіти.

Отже, майбутнім фахівцям у сфері геодезії ще на етапі навчання необхідно

розвивати цифрові компетентності. Додаткові можливості для цього надає вивчення мов програмування. Для здобувачів геодезичної освіти рекомендується почати з простої в освоєнні та потужної мови програмування — Python, опис можливостей і сфер застосування якої наведено в наступних розділах.

Розділ 2. АНАЛІЗ ПЕРСПЕКТИВ, МОЖЛИВОСТЕЙ І ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ МОВОЮ PYTHON

2.1 Мова програмування Python та історія її створення

На сьогодні багато контрибуторів зробили свій внесок у розвиток мови програмування Python, але заслужену славу творця цієї перспективної та популярної мови отримав Гвідо ван Россум (Guido van Rossum). Робота над Python розпочалася ще у 1980-х роках, і перша версія мови була випущена в 1991 році.

Щодо назви мови — Python, формально вона походить від назви рептилії, тому як логотип обрано зображення змії з родини пітонів. Проте насправді назва «Python» відсилає не до змії, а до британського гумористичного шоу Террі Гіллама — «Летючий цирк Монті Пайтона» [19], яке вперше вийшло в ефір 5 жовтня 1969 року і включало 45 епізодів, об'єднаних у чотири сезони. Шоу складалося з окремих скетчів і анімацій, без єдиного сюжету [20].



Рис. 2.1. Логотип мови програмування Python [20]

Тепер перейдемо до визначення мови Python. Python — це популярна, високорівнева мова програмування загального призначення з динамічною типізацією, орієнтована на підвищення продуктивності розробки програмних продуктів. Python працює на більшості відомих платформ — Linux, Windows, MacOS. Вона є відкритим та вільним програмним забезпеченням, виконується

шляхом інтерпретації байт-коду і підтримує кілька парадигм програмування: функціональне та об'єктно-орієнтоване.

Код на Python компактний і легкий для читання (рис. 2.2). Мова характеризується динамічною типізацією, наявністю зручних структур даних (кортежі, списки, словники, множини), великою стандартною бібліотекою та широким спектром сторонніх бібліотек різного призначення. Інтерпретатор Python має інтерактивний режим роботи, що дозволяє виконувати код безпосередньо з консолі та одразу отримувати результат (рис. 2.3).



Рис. 2.2. Переваги мови Python [21]

Завдяки цим перевагам Python широко застосовується прикладними програмістами, зокрема інженерами та науковцями [21].

```
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 1
>>> b = 2
>>> a + b
3
>>> █
```

Рис. 2.3. Інтерпретатор Python

Python — досить потужний, щоб вирішувати широкий спектр завдань. Хоча комп'ютери ще не настільки "розумні", щоб виконувати все за нас, інструменти Python є надзвичайно універсальними та можуть використовуватися в різних проєктах.

Можливо, ви з острахом дивитеся на комп'ютерних фахівців, побоюючись, що ніколи не зможете досягти їхнього рівня майстерності. Однак Python ідеально підходить для новачків, адже він не відлякує надміром фігурних чи квадратних дужок, крапок з комами чи двокрапок. По суті, синтаксис Python дуже схожий на англійську мову. Щоб продемонструвати це, порівняємо програми, написані різними мовами програмування.

Серед програмістів існує традиція: перший застосунок або ж програма на новій мові мусить виводити на екрані слова "Hello, world!". Програма для цього на Python виглядає так:

```
>>>print("Hello, world!")
```

Цей код простий і зрозумілий для кожного, хто хоча б трохи знає англійську. Як і в усіх мовах програмування, тут є дужки та лапки, але до них доведеться звикнути, якщо плануєте займатися програмуванням серйозно.

Для новачків існують й інші мови програмування. Наприклад, Ruby, де код програми також можна написати в один рядок, хоча сама команда виглядає дещо менш інтуїтивно:

```
>>>puts "Hello, world!"
```

У JavaScript такого результату можна досягти двома способами, які можуть виглядати менш зрозуміло для новачків. Вже на перших етапах вивчення JavaScript доведеться навчитися розрізняти консольний вивід і спливаюче вікно:

- 1) `>>>console.log ('Hello, world!');`
- 2) `>>>alert ('Hello, world!');`

Кожен із цих способів має свої особливості використання, що додає JavaScript додаткового рівня складності для новачків.

Якщо подивитися на "гігантів" програмування, таких як Java, C та C++, стає очевидним, що для найпростішої програми потрібно використовувати загадкові команди та велику кількість фігурних дужок (табл. 2.1). Наприклад:

Java	C	C++
<pre>public class Main { public static void main(String[] args) { System.out.println("Hello, World!"); } }</pre>	<pre>#include <stdio.h> int main() { printf("Hello, world!\n"); }</pre>	<pre>#include <iostream> int main() { std::cout << "Hello, world!"; }</pre>

Таблиця 2.1. Порівняння синтаксису мов програмування Java, C та C++

Просте порівняння цих програм одразу показує елегантність і лаконічність Python як мови програмування. Python дозволяє зосередитися на логіці роботи програми замість того, щоб боротися з правилами розстановки розділових знаків. Саме тому 80% провідних університетів і шкіл світу обирають Python для вступних курсів з програмування. Хоча розмір програми та використання зрозумілих команд не є визначальними факторами її якості, ці особливості допомагають початківцям уникнути розгубленості на перших заняттях і

зберегти мотивацію до вивчення. Опанувавши базові команди, новачки можуть упевнено переходити до складніших конструкцій та глибшого вивчення програмування [22].

Серед небагатьох недоліків Python, на мою думку, можна виділити порівняно невисоку швидкість виконання програм, що зумовлено його інтерпретованою природою. Проте цей недолік цілком компенсується численними перевагами мови, особливо у випадках, коли програми не потребують високої швидкодії.

2.2 Можливості застосування, перспективи та способи вивчення

Станом на сьогодні, мова програмування Python є однією з найзатребуваніших у світі. Вона посідає перше місце в рейтингу PYPL та стала "мовою року" за версією ТЮВЕ. Крім того, за даними GitHub, Python є найбільш використовуваною мовою програмування за четвертий квартал 2024 року. Сфера застосування Python надзвичайно широка:

- **Веб-розробка:** завдяки фреймворкам, таким як Django та Flask, Python активно використовується для створення веб-додатків.
- **Аналіз даних та машинне навчання:** бібліотеки, як-от Pandas, NumPy та Scikit-learn, роблять Python основним інструментом у цій галузі.
- **Автоматизація:** Python застосовується для автоматизації рутинних завдань, що підвищує ефективність роботи.
- **Розробка ігор:** за допомогою бібліотек, таких як Pygame, можна створювати прості ігри.
- **Наукові дослідження:** Python використовується в наукових обчисленнях та моделюванні.
- **Розробка програмного забезпечення:** мова застосовується для створення різноманітних програмних продуктів.
- **Освіта:** завдяки простоті синтаксису, Python часто використовується для навчання програмуванню.

Згідно з дослідженням JetBrains, Python продовжує утримувати лідерські позиції серед мов програмування, що підтверджує його популярність та широке застосування у різних сферах.

17% – аналіз даних

27% – веб-розробка

11% – машинне навчання

11% – адміністрування

34% - DevOps скрипти

Python неодноразово довів свою ефективність не тільки як мова для навчання, але й як інструмент для вирішення широкого кола практичних завдань. Проте вивчення будь-якої мови програмування, зокрема Python, може викликати певні закономірні побоювання щодо її рівня абстракції, системи типізації, а також наявності бібліотек зі стандартними алгоритмами.

Python, будучи мовою високого рівня з динамічною типізацією, має величезну стандартну бібліотеку та доступ до численних зовнішніх бібліотек. Це забезпечує готові рішення для більшості задач, що допомагає початківцям і професіоналам швидко знаходити оптимальні підходи, не витрачаючи багато часу на реалізацію базових алгоритмів з нуля.

Справді, сфери застосування Python вражають! (рис. 2.4). Ця мова є потужною завдяки наявності численних бібліотек, створених для вирішення завдань у різних галузях, що значно розширює можливості чистого Python. Наприклад, для наукових обчислень є бібліотеки **NumPy** і **SciPy**. Якщо вас цікавить розробка ігор, ідеальним інструментом буде **Pygame**. Для обробки природної мови та розробки штучного інтелекту корисною стане бібліотека **NLTK** (Natural Language Toolkit).

Таким чином, бібліотеки Python застосовуються у широкому діапазоні галузей: астрономії, криптографії, мультимедійних даних, біології, мережеских технологіях, веб-розробці та багатьох інших. Це робить Python універсальним інструментом для вирішення різноманітних завдань.



Рис. 2.4. Можливості застосування мови програмування Python [22]

Якщо цей світ нескінченних можливостей ще не переконав вас у необхідності вивчення Python, варто знати, що багато відомих компаній активно використовують цю мову. Наприклад, Disney застосовує Python у своєму виробничому процесі, а Mozilla випускає численні пакети з відкритим кодом, створені на основі Python. Мова використовується для загального програмування, автоматизації, тестування, а також у стартапах, розробці ігор та веб-середовищах.

Python також широко застосовується такими компаніями, як NASA, Dropbox і YouTube. Facebook, Instagram і Prezi працюють на платформі Django, яка підтримує роботу їхніх серверів і забезпечує функціональність веб-додатків. Крім того, як зазначалося в попередніх розділах, Python є потужним інструментом для розширення функціональності QGIS та автоматизації геоінформаційних процесів. Інтеграція Python у QGIS дозволяє користувачам створювати скрипти, розробляти плагіни та налаштовувати програму відповідно до конкретних потреб.

Підсумовуючи вищесказане, можна дійти до висновків, що Python має низький поріг, високу стелю та широкі стіни, чудово підходячи як для початківців, так і для професійних програмістів для створення веб-проектів, побудови ігор, розв'язання математичних задач чи побудови графіків із своїм чітким та послідовним синтаксисом, написання плагінів та розширень для різноманітних застосунків, що чудово дає старт для вступу до захоплюючої подорожі у світ програмування. До того ж набуття навичок у програмуванні дає суттєві переваги – тренує розум, що, в свою чергу, позитивно впливає на здатність логічно, творчо та критично мислити. До переваг також належать: уміння вирішувати проблеми; активізація процесів в обох півкулях головного мозку; розвиток уміння створювати власні алгоритми вирішення проблем; набуття впевненості під час створення нового.

Отже, Python є перспективною мовою програмування, яка знаходить широке застосування серед програмістів, інженерів і науковців. На початку вивчення будь-якої мови програмування процес може здаватися складним, лякаючим і незрозумілим. Проте, на мою думку, Python — одна з найпростіших для опанування мов, хоча іноді можуть виникати певні труднощі. Досить часто під час її вивчення я стикався з фрагментами коду в Інтернеті, які залишалися для мене незрозумілими.

```

1  from django.http import HttpResponse, HttpRequest
2
3
4  def index(request: HttpRequest) -> HttpResponse:
5      print(request)
6      return HttpResponse("Hello, world.")
7

```

Рис. 2.5. Фрагмент коду із бібліотеки Python - Django

Код, наведений на рис. 2.5, взято з документації Django — популярного фреймворку на Python для розробки веб-сайтів. Досвідчені розробники легко скажуть, що в ньому немає нічого складного, але навіть кілька простих на вигляд рядків коду можуть здатися дуже незрозумілими для початківця. Наприклад:

- Чому є рядки з відступом?
- Що та для чого `django.http`?
- Чому є елементи, які знаходяться у дужках?

Зрозуміти, як усе працює, коли ви ще не знаєте Python, дуже непросто. Щоб створювати програми, які вас цікавлять, необхідно освоїти основні "будівельні блоки" цієї мови програмування. Лише з розумінням цих базових компонентів можна буде впевнено розробляти функціональні й цікаві проєкти.

Більшість туторіалів (покрокових інструкцій у форматі відео або зображень) передбачають, що спершу ви повинні вивчити весь синтаксис мови, а вже потім почнете створювати щось цікаве. Однак ви можете поєднувати вивчення базових елементів (core) з розробкою тих проєктів, які вас цікавлять. Далі я наведу етапи вивчення Python, які допомогли мені в моєму навчанні:

1. Потрібно зрозуміти, що вас мотивує

Перш ніж починати вивчення чого-небудь, не лише Python, важливо запитати себе, чому саме ви хочете це освоїти. Адже попереду вас чекає довга та подекуди непроста подорож. Без достатньої мотивації досягти бажаного буде

складно, оскільки саме мотивація допоможе вам зрозуміти, чого саме ви прагнете досягти й надасть сили йти вперед, незважаючи на труднощі.

2. Необхідно вибрати галузь, яка цікавить, наприклад:

- Наука про дані / Машинне навчання
- Веб-сайти
- Ігри
- Мобільні застосунки
- Скрипти для автоматизації вашої роботи
- Апаратні засоби / Сенсори / Роботи

Варто обрати одну або дві галузі, які вас цікавлять, і спробувати розвиватися саме в цих напрямках. Якщо ви ще не визначилися з вибором, я б рекомендував приділити трохи часу дослідженню кожної з галузей, а потім зупинитися на тій, що приносить вам найбільше задоволення. Це допоможе вам зосередитися на тому, що дійсно цікаве, й отримувати задоволення від процесу навчання.

3. Основний синтаксис

Цей процес надзвичайно важливий, оскільки вам необхідно знати базовий синтаксис Python, перш ніж глибше занурюватися в обрану галузь. Особисто я вивчав це за допомогою книг, туторіалів на YouTube та онлайн-курсів. Нижче я наведу кілька корисних ресурсів для вивчення:

- книга «A Bite of Python» автора Swaroop С. Н. – навчає концепції, з основ до більш складних програм;
- «Грохаємо алгоритми» автора Адітья Бхаргава – книга, яка навчить вас алгоритмізувати задачі та дасть уявлення про загальні алгоритми;
- «The Python Tutorial» - офіційна документація.

4. Розробляти структуровані продукти

Після засвоєння базового синтаксису можна починати розробку власних проєктів. Це один із найкращих способів навчитися, адже практика допомагає закріпити теоретичні знання. Працюючи над проєктами, ви не лише дізнаєтеся багато нового, але й поступово сформуєте портфоліо, яке стане в пригоді при пошуку стажування чи роботи.

Однак "вільні" проєкти можуть стати викликом, оскільки багато речей будуть вам незрозумілими. Тому я рекомендую обирати більш структуровані проєкти, над якими ви зможете поступово відчувати, що вже маєте достатньо знань для створення власних ідей. Приклади й теми таких проєктів завжди можна знайти в Інтернеті.

Також важливо навчитися шукати рішення ваших проблем в Інтернеті. Під час навчання ви зустрінете багато нового та незрозумілого, а в процесі розробки власних програм виникатиме безліч питань і багів у коді. Тому надзвичайно важливо вміти аналізувати та знаходити рішення помилок за допомогою ресурсів онлайн.

Більшість помилок у своєму коді ви зможете знайти та вирішити, використовуючи веб-ресурс StackOverflow — сайт, де люди описують свої проблеми в програмуванні, а досвідчені фахівці або ті, хто вже стикався з подібними ситуаціями, надають відповіді. Також не забувайте, що Google — ваш найкращий друг у вирішенні будь-яких проблем із кодом.

5. Ускладнюйте продукти, над якими ви працюєте

Необхідно збільшувати рівень складності і обсяг ваших проєктів. Якщо ви вже запевняєте себе у тому, що те, що ви зараз робите, то ймовірно це означає, що настав час спробувати щось нове та складніше [26]?

2.3 Python як інструмент для геодезиста

Python відіграє важливу роль у QGIS та геодезії, забезпечуючи потужні інструменти для автоматизації, аналізу даних і розширення функціональності програмного забезпечення. Ось основні аспекти застосування Python у цій сфері:

- Автоматизація процесів
 - Скрипти для обробки даних: Python дозволяє створювати скрипти, які автоматизують рутинні задачі, такі як імпорт, експорт та обробка геоданих.
 - Пакетні обробки: Можна автоматизувати обробку великої кількості файлів, що суттєво економить час.
 - Обробка геопросторових даних: Бібліотеки, такі як GeoPandas та Shapely, дозволяють ефективно працювати з векторними даними, виконувати просторові операції та аналіз.
- Аналіз даних

- Геостатистичний аналіз: Використовуючи бібліотеки, такі як GeoPandas, можна проводити складний аналіз геопросторових даних.

- Обробка просторових даних: Python забезпечує доступ до потужних інструментів для маніпуляції просторовими даними, включаючи фільтрацію, агрегацію та візуалізацію.

- Використання бібліотек Matplotlib та Seaborn сприяє створенню інформативних графіків та карт, що допомагає в інтерпретації геодезичних даних.

- Розробка плагінів

- Створення плагінів для QGIS: Python дозволяє створювати плагіни, які додають нові функції та інструменти до QGIS, що покращує робочий процес.

- Налаштування інтерфейсу: Плагіни можуть також змінювати або покращувати інтерфейс програми, підлаштовуючи його під специфічні потреби користувача.

- Взаємодія з API

- Інтеграція з веб-сервісами: Python може взаємодіяти з різними API, що дозволяє отримувати та використовувати зовнішні геоінформаційні дані, такі як дані з WMS або WFS сервісів.

- Збирання даних з інтернету: З його допомогою можна автоматично збирати дані з веб-сайтів, що значно полегшує процес збору інформації.

- Візуалізація даних

- Графічне представлення: Python надає можливості для створення графіків та карт для візуалізації даних, що допомагає в аналізі та прийнятті рішень.

- Спеціалізовані інструменти для візуалізації: Використання бібліотек, таких як Matplotlib або Seaborn, для створення візуальних звітів і графіків.

Перспективи вивчення Python для геодезистів:

- Підвищення конкурентоспроможності: Володіння Python стає важливою навичкою на ринку праці, оскільки все більше компаній впроваджують автоматизацію та аналіз даних у свої процеси.

- Розширення професійних можливостей: Знання Python відкриває доступ

Розділ 3. ВИКОРИСТАННЯ PYTHON В СТВОРЕННІ ПЛАГІНУ ДЛЯ ПРОГРАМНОГО ПРОДУКТУ QGIS

3.1 Мета, ціль та опис функціональності розробки

Для покращення цифрової компетентності студентів у сфері геодезії активно використовуються інноваційні технології та інструменти, такі як плагіни для QGIS, написані на Python. Це надає можливість не лише автоматизувати рутинні процеси, але й формувати навички роботи з сучасними цифровими рішеннями, що мають практичне значення для їхньої майбутньої професії.

Плагін для QGIS, розроблений на Python, виконує важливу функцію автоматизації задачі лабораторних робіт, що значно покращує зручність як для студентів, так і для викладачів. У процесі роботи над лабораторними завданнями студенти мають змогу моделювати реальні ситуації, коли результати їхньої праці миттєво передаються викладачеві, мінімізуючи потребу в додаткових діях та суттєво скорочуючи час, який зазвичай витрачається на задачу робіт. Такий підхід сприяє розвитку вміння працювати в команді, організовувати власний робочий процес і використовувати сучасні технології для досягнення результатів.

Основні особливості функціонування плагіну включають:

1. Інтеграція в середовище QGIS – Плагін гармонійно поєднується з інтерфейсом та функціями QGIS, забезпечуючи зручний доступ до автоматизованого процесу передачі даних. Коли студент завершує завдання і натискає команду збереження, плагін автоматично активує процедуру передачі файлу.

2. Автоматизована передача результатів через Telegram-бот – Використовуючи API Telegram, плагін дозволяє миттєво передавати збережений файл до викладача через спеціально створений бот. Це забезпечує оперативний контроль за ходом виконання робіт, дозволяючи викладачеві у режимі реального часу отримувати результати студентів та оцінювати їхню активність.

3. Надійність і зручність передачі даних – Плагін значно зменшує ризик втрати чи неправильної передачі файлу. Студенти можуть бути впевнені, що їхня робота дійде до викладача у тому вигляді, у якому вони її здали. Це створює безпечне і організоване середовище для контролю над процесом здачі завдань.

4. Функція логування – Плагін фіксує події збереження для кожного студента, що включає реєстрацію дати і часу останнього збереження файлу. Це не лише додає додаткову інформативність, а й дозволяє відслідковувати послідовність виконання завдань. Такий підхід підвищує прозорість та дає змогу оперативно реагувати на затримки чи інші проблеми у виконанні завдань.

5. Гнучкість і подальший розвиток – Використання Python як основної мови для написання плагіну відкриває широкі можливості для подальшого вдосконалення інструменту. Плагін може бути адаптований під різні навчальні процеси, а також доповнений новими функціями, що дозволить розширити його можливості. Python забезпечує гнучкість у додаванні нових можливостей, таких як підтримка інших месенджерів, налаштування різних типів повідомлень для конкретних завдань або створення більш розширених звітів.

Отже, впровадження цього плагіну у навчальний процес у сфері геодезії дозволяє ефективно оптимізувати час, присвячений на контроль і здачу робіт, забезпечує викладачів зручними інструментами для аналізу, а також сприяє формуванню у студентів навичок роботи з передовими технологіями. Водночас, плагін можна адаптувати для використання в інших дисциплінах, де необхідний швидкий та надійний обмін результатами між студентами та викладачами.

3.2 Вибір архітектури та технологічного стеку плагіну

Для реалізації плагіну QGIS, який автоматизує процес задачі лабораторних робіт, використовуються наступні сучасні технології та інструменти:

1. Python – основна мова програмування для написання плагіну. Python надає розробникам інтуїтивно зрозумілий синтаксис і широкий набір бібліотек для різних задач, що значно полегшує створення і підтримку плагіну.

a. PyMongo – бібліотека для взаємодії з базою даних MongoDB. Вона дозволяє швидко додавати, оновлювати і читати дані з MongoDB, що забезпечує збереження та управління файлами з результатами робіт студентів [37].

b. pyTelegramBot (telebot) – бібліотека для інтеграції з Telegram API, яка дозволяє надсилати повідомлення, файли та керувати ботом зручним чином. Вона необхідна для передачі збережених робіт до викладача через Telegram-бот [38].

c. Внутрішні бібліотеки QGIS – набір інструментів, що дозволяють інтегруватися з функціоналом QGIS, наприклад, обробляти події, пов'язані з збереженням файлів, взаємодіяти з інтерфейсом і забезпечувати логіку плагіну [39].

d. FastAPI – веб-фреймворк для створення API, який тут використовується для встановлення webhook для Telegram-бота. Це дозволяє Telegram-боту в режимі реального часу отримувати повідомлення про нові завантаження робіт [40].

e. MongoDB – не-реляційна база даних, яка використовується для збереження файлів, пов'язаних із лабораторними роботами студентів. MongoDB зберігає інформацію у вигляді документів, що дозволяє гнучко організувати дані і легко масштабувати систему у майбутньому [41].

2. Docker – платформа для контейнеризації додатків. Docker використовується для розгортання MongoDB на сервері разом із Telegram-ботом, що дозволяє легко налаштувати та керувати середовищем. Контейнеризація забезпечує високу мобільність та ізоляцію сервісів.

3. ngrok – інструмент для створення тунелів і прокидання локального

сервера в інтернет. Використовується для тимчасового доступу до локально розгорнутого FastAPI додатка (вебхуку) під час розробки, що дозволяє Telegram-боту отримувати дані з локального сервера.

Також було вирішено використати триланкову структуру проекту (рис. 3.1), що складається з наступних компонентів:

- сервер;
- клієнт;
- база даних;



Рис. 3.1. Триланкова структура ПЗ (програмного забезпечення)

У зазначеній схемі ключову роль відіграє сервер, який централізує базову бізнес-логіку та надає клієнту доступ до бази даних. Для взаємодії з користувачем використовується клієнтський додаток Telegram. Користувальницький інтерфейс обробляє інформацію перед передачею її на серверну частину, а потім повертає результат обробки даних користувачеві.

Рівень бази даних також має важливе значення для роботи платформи, оскільки він не лише забезпечує доступ до даних, але й гарантує їх цілісність. Крім того, на цьому рівні зазвичай реалізується бізнес-логіка, яка взаємодіє виключно з базою даних, без залучення зовнішніх джерел.

3.3 Встановлення плагіну та ілюстрація його можливостей

Для ілюстрації взаємодії користувача з інтерфейсом використаємо діаграму прецедентів UML (рис. 3.2) — схему, яка відображає взаємозв'язок між учасниками та прецедентами в системі. Її також називають діаграмою

використання.

Діаграма прецедентів складається з таких елементів: учасників, прецедентів (варіантів використання), обмежених системною межею (прямокутником), асоціацій між учасниками та прецедентами, а також взаємозв'язків між прецедентами і між учасниками [42].

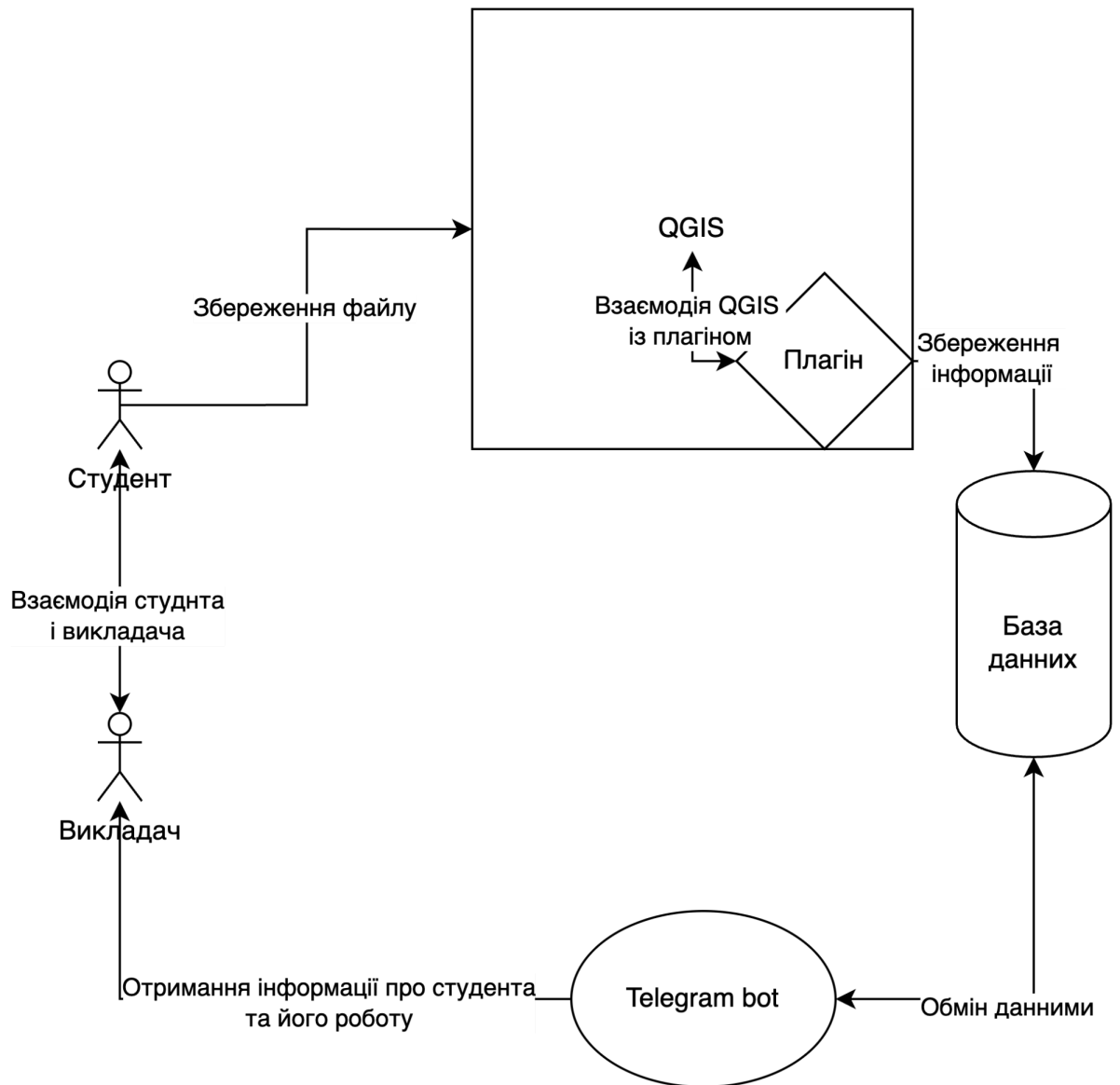


Рис. 3.2. UML діаграма прецедентів системи

Для безпосередньої роботи плагіну локально виконати наступні кроки.

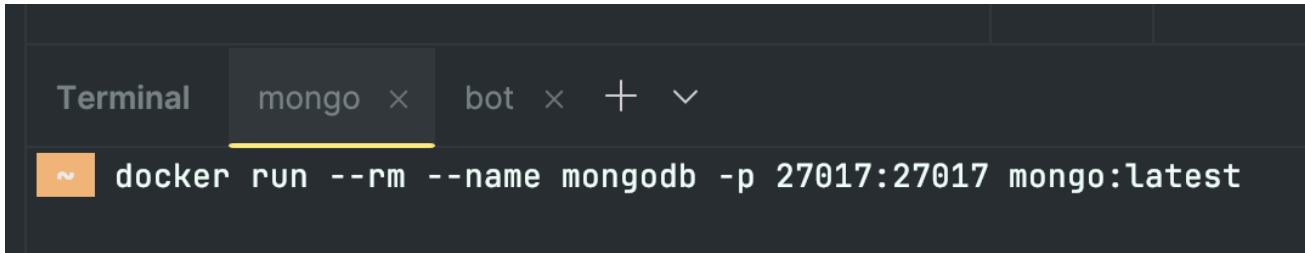
1) Розгортання MongoDB за допомогою Docker

Запустити MongoDB у контейнері Docker, щоб забезпечити доступ до бази

даних для зберігання та обробки файлів, які студенти надсилають через плагін.

Використовуйте наступну команду:

```
docker run --rm --name <container-name> -p 27017:27017 mongo:<version>
```

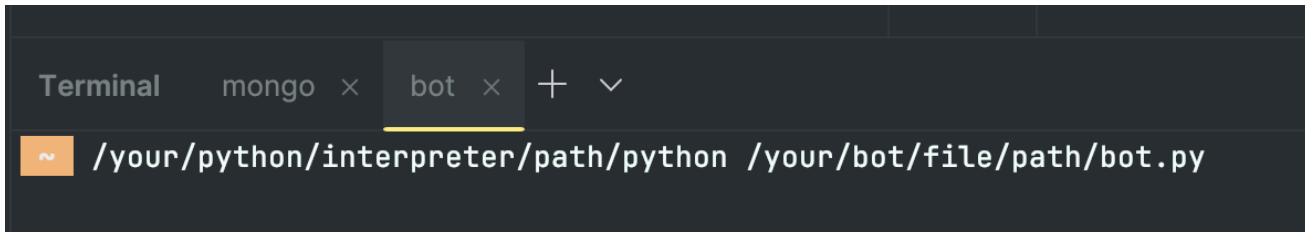

 A screenshot of a terminal window with a dark background. The title bar shows 'Terminal', 'mongo x', 'bot x', and a '+' icon. The command entered is 'docker run --rm --name mongodb -p 27017:27017 mongo:latest'. The prompt is a tilde '~'.


```
Terminal  mongo x  bot x  +  v
~ docker run --rm --name mongodb -p 27017:27017 mongo:latest
```

Рис. 3.3. Команда запуску MongoDB в Docker контейнері

2) Запуск Телеграм-бота

Після налаштування MongoDB перейдіть до запуску Телеграм-бота, який отримуватиме файли від плагіну. Бот повинен бути налаштований для взаємодії з MongoDB (через бібліотеку PyMongo) та приймати дані через webhook, створений за допомогою FastAPI. Запустіть скрипт бота, забезпечивши доступ до MongoDB.


 A screenshot of a terminal window with a dark background. The title bar shows 'Terminal', 'mongo x', 'bot x', and a '+' icon. The command entered is '/your/python/interpreter/path/python /your/bot/file/path/bot.py'. The prompt is a tilde '~'.


```
Terminal  mongo x  bot x  +  v
~ /your/python/interpreter/path/python /your/bot/file/path/bot.py
```

Рис. 3.4. Команда запуску Телеграм бота

3) Реєстрація в Телеграм боті

Щоб підключитися до бота, необхідно відкрити застосунок Телеграм та знайти бота за його ім'ям *@qgis_telegram_logging_plugin_bot* (див. рис. 3.5)

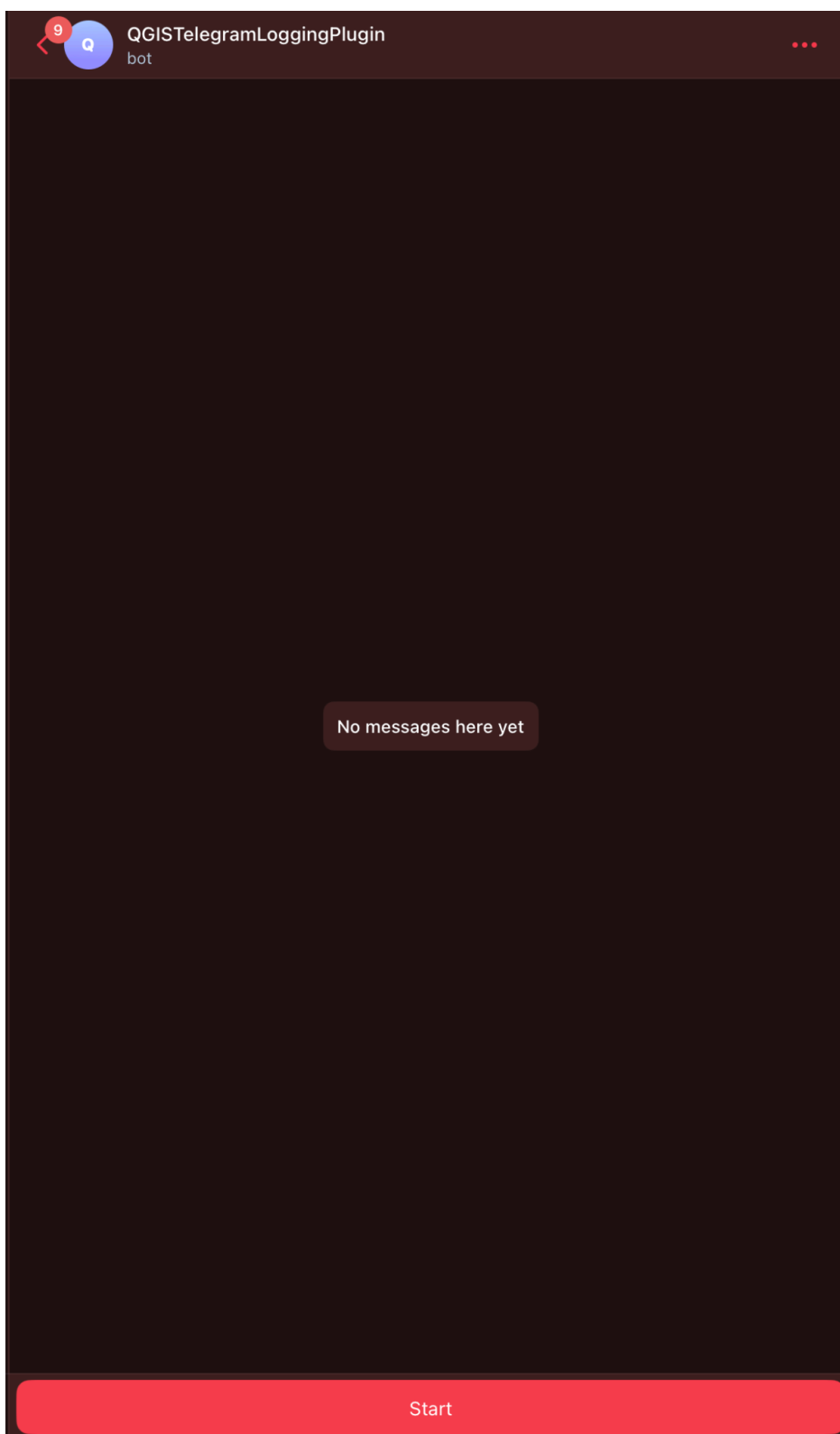


Рис. 3.5. Стартове меню бота

Для того щоб розпочати із ним роботу, необхідно натиснути кнопку Start, або ж написати ботові /start, після чого він повідомляє вас про подальші кроки реєстрації (див. рис. 3.6)

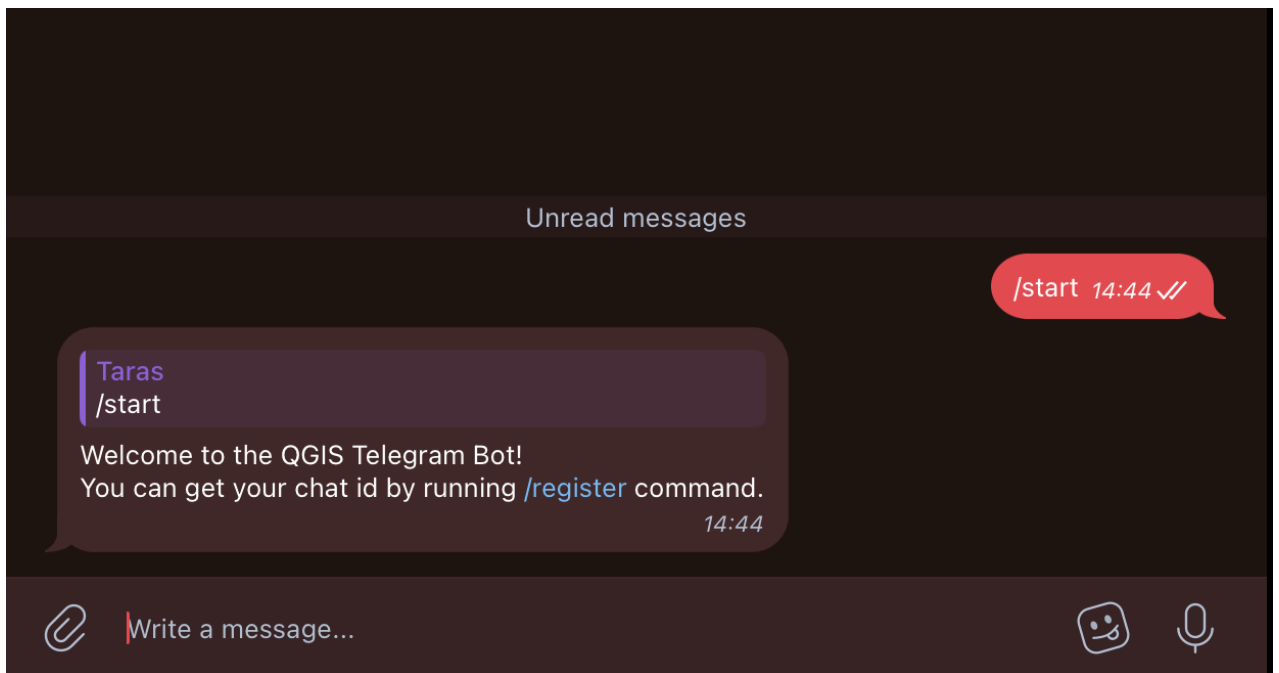


Рис. 3.6. Опрацювання команди /start

Наступною командою для бота є `/register` після виконання якої бот попросить вас поділитися вашими контактними даними (кнопка “Share your contact for registration” внизу) (див. рис. 3.7), після чого повідомить вас про успішну реєстрацію, та надасть вам інформацію про ідентифікатор вашого чату (див. рис. 3.8), який знадобиться пізніше при під’єднанні студента до плагіну (див. рис. 3.14)

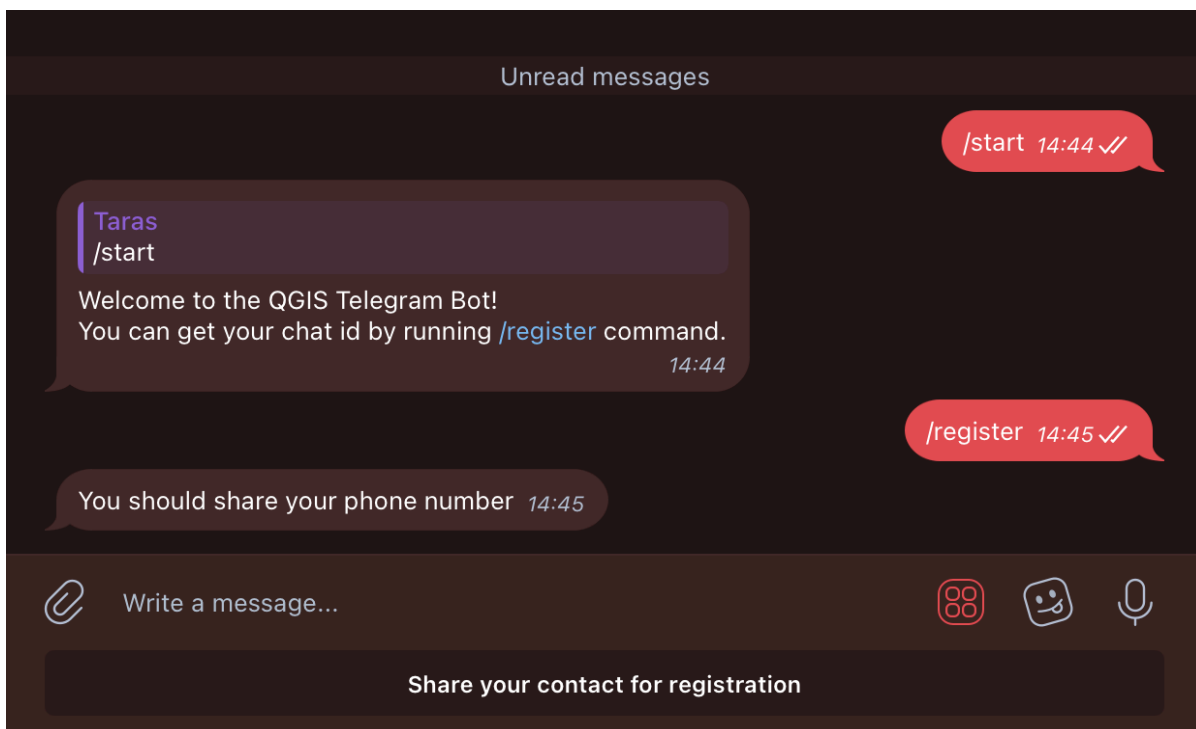


Рис. 3.7. Опрацювання команди /register

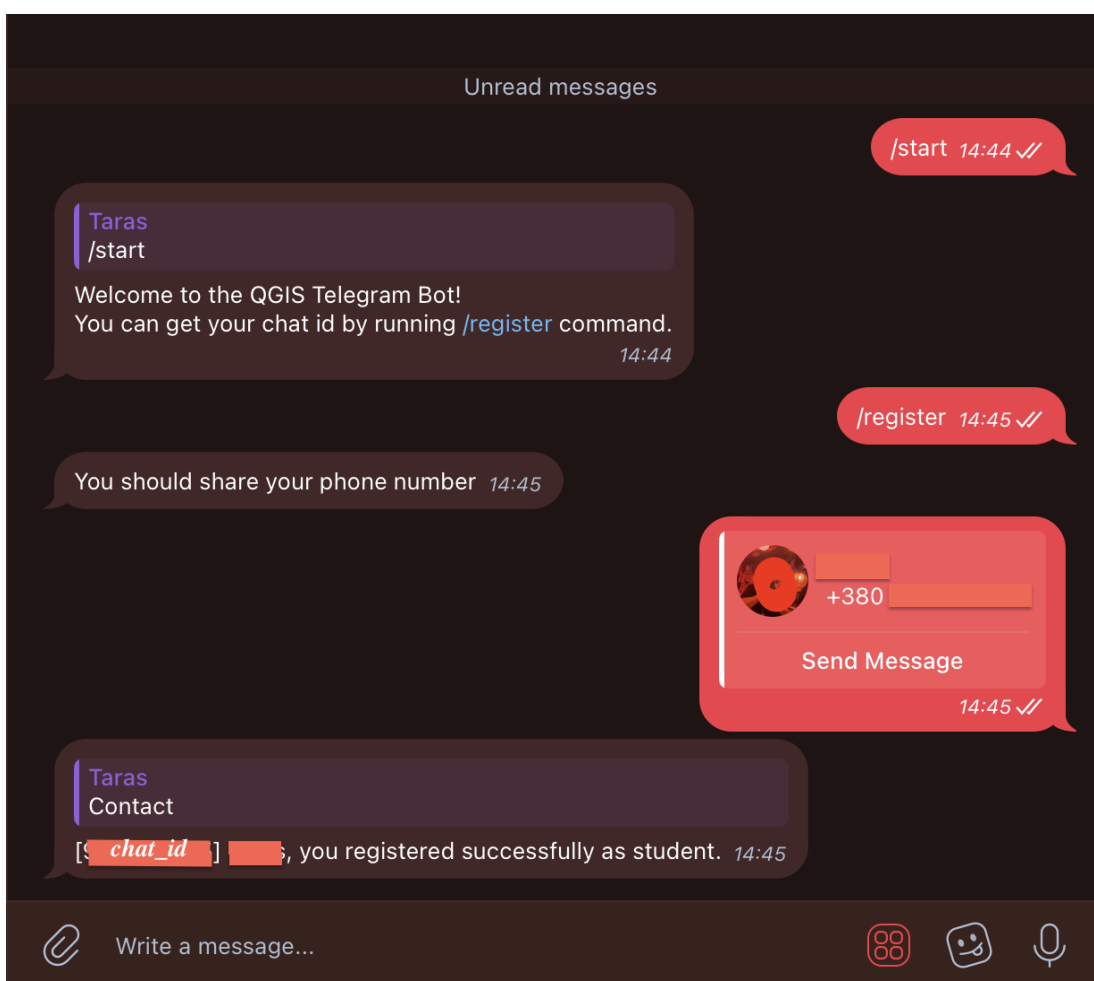


Рис. 3.8. Опрацювання вхідного контакту

Якщо ви студент, то на цьому ваша реєстрація в боті закінчена, якщо ж ви викладач, вам необхідно продовжити командою `/as_teacher`, яка переведе тип вашого користувача як “Викладач”. Після введення команди, бот попросить вас пароль. У випадку хибного, бот вас просто про це повідомить (див. рис. 3.9), а якщо ж пароль вірний, то зміниться також тип користувача (див. рис. 3.10). Цей крок був необхідний для того, щоб плагін знав кому відправляти роботу студента.

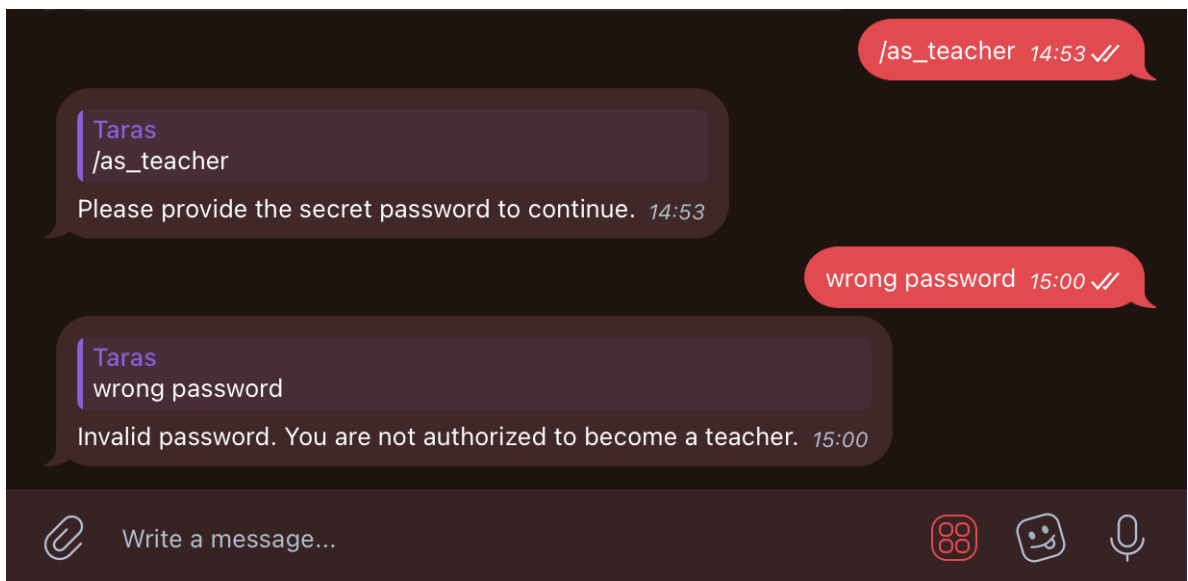


Рис. 3.9. Зміна типу користувача на “Викладач” (хибний пароль)

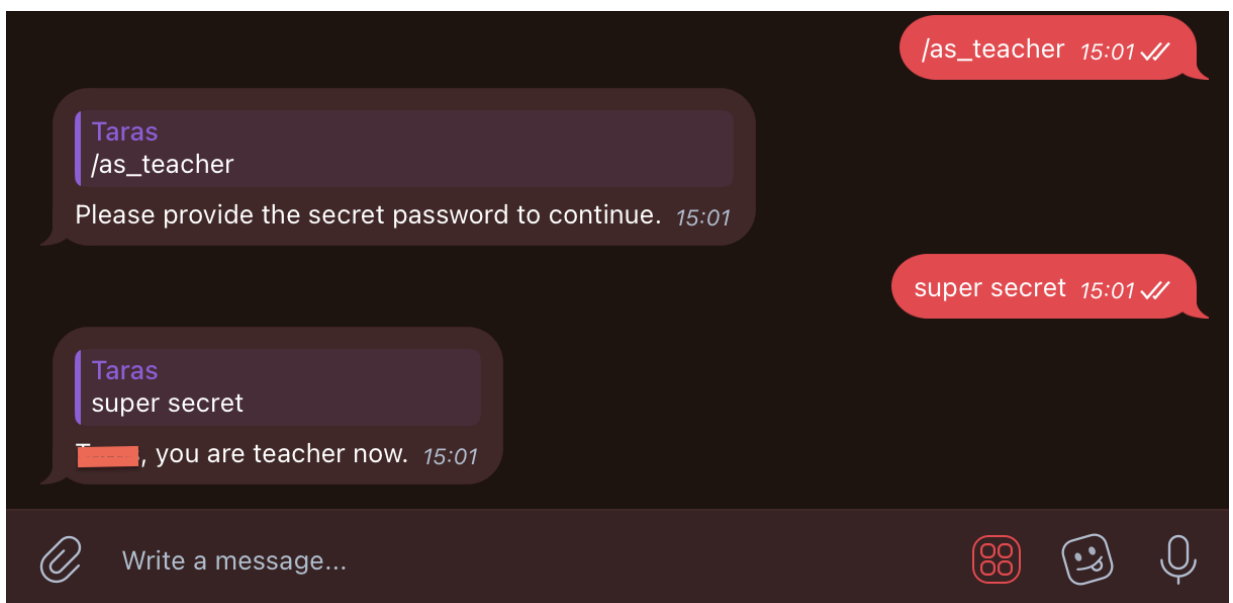


Рис. 3.10. Зміна типу користувача на “Викладач” (вірний пароль)

Викладач також має можливість переглянути список зареєстрованих студентів командою `/students` (див. рис. 3.11).

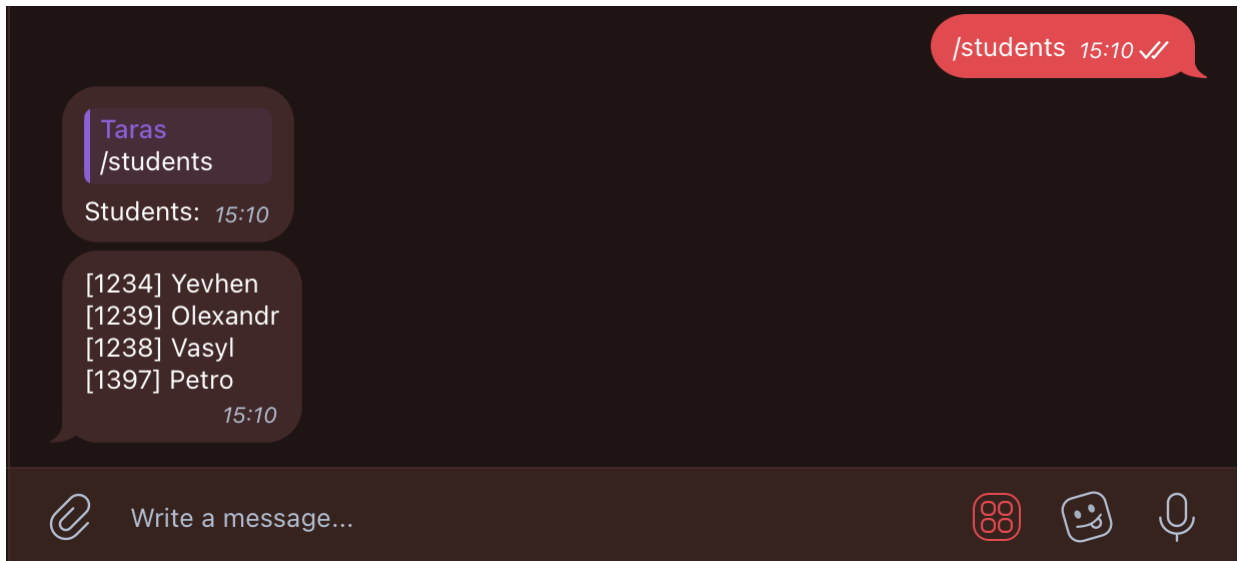


Рис. 3.11. Отримання списку зареєстрованих студентів

4) Ініціалізація об'єкта плагіну в QGIS Python консолі

Для підключення плагіну необхідно відкрити програму QGIS та в Python консолі ввести наступну команду `exec(open('/path/to/plugin.py').read())`. Для студента необхідно вказати його ідентифікатор чату отриманий на наступному кроці командою `telegram_gis.student = chat_id` (див. рис. 3.12).

Якщо ж викладач або студент пропустив минулі кроки, але підключив плагін командою `exec(open('/path/to/plugin.py').read())`, то при збереженні файлу в вікні програми QGIS буде повідомлення, яке вкаже викладачу (див. рис. 3.13) або студенту (див. рис. 3.14) про його наступні кроки.

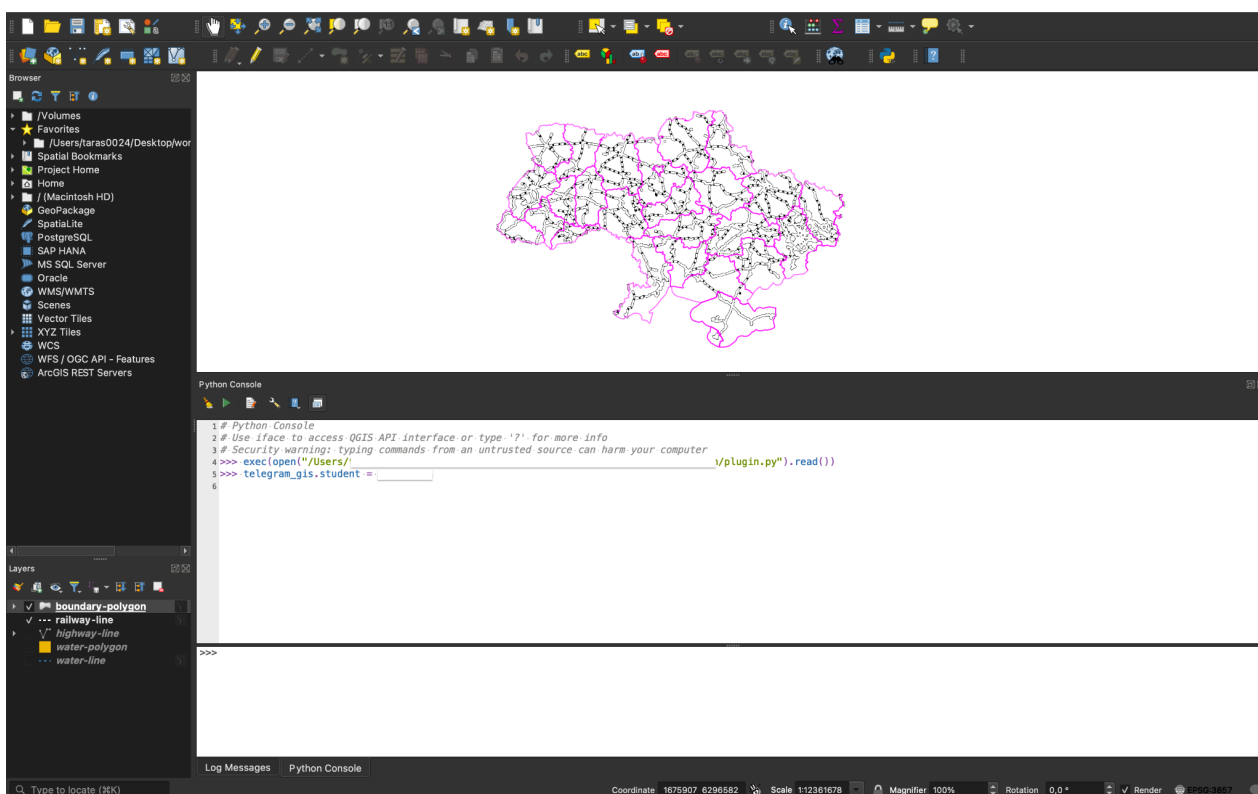


Рис. 3.12. Підключення плагіну

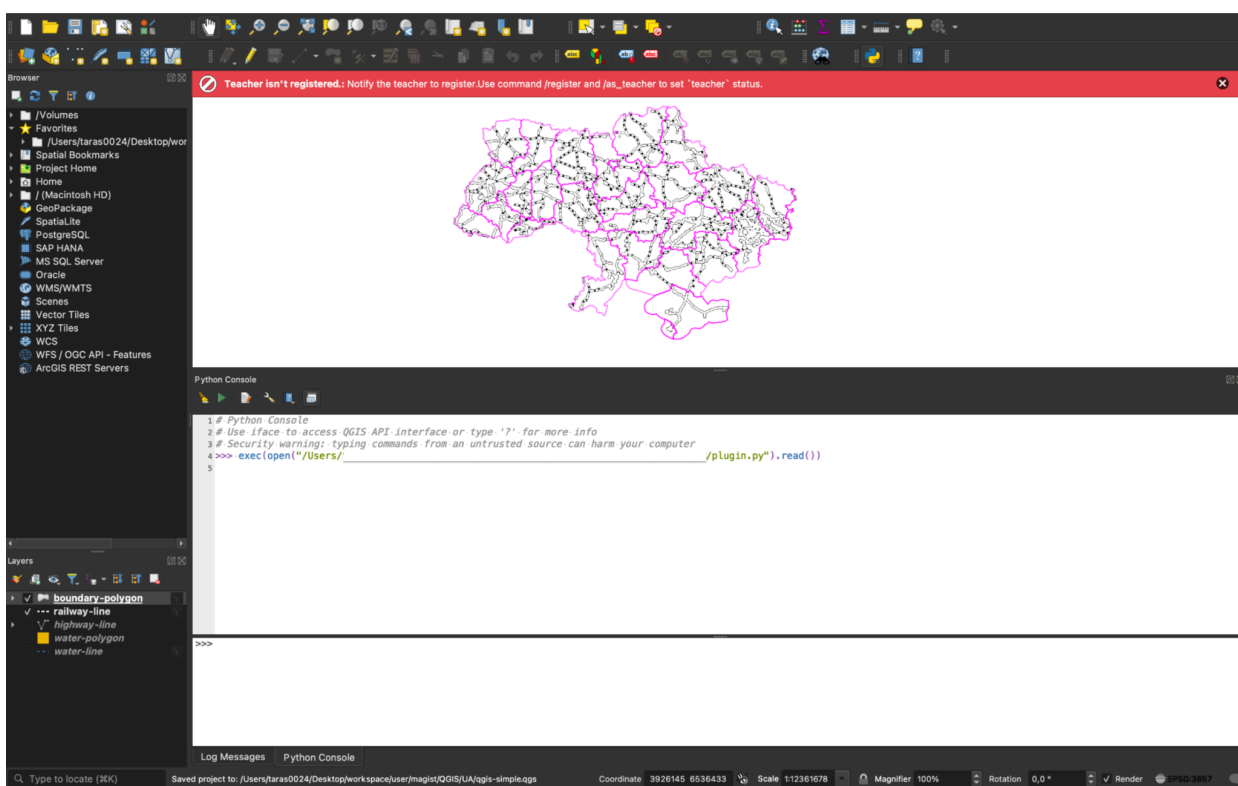


Рис. 3.13. Повідомлення про незареєстрованого викладача

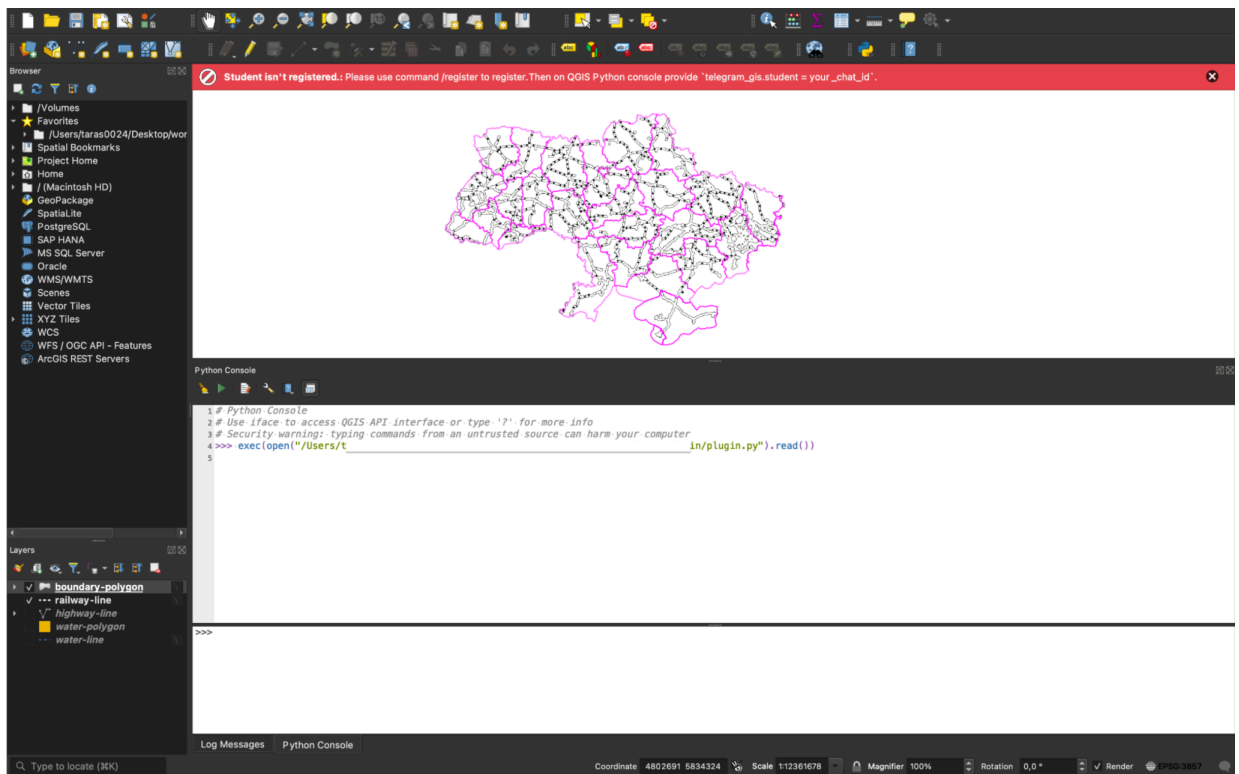


Рис. 3.14. Повідомлення про незареєстрованого студента

5) Збереження файлу в QGIS

Після налаштування всіх попередніх кроків студента і викладача, при збереженні студентом файлу буде відправлятися повідомлення викладачеві (див. рис. 3.15) про успішну здачу студентом роботи та сама робота, а студента проінформує кому було відправлено його роботу (див. рис. 3.16)

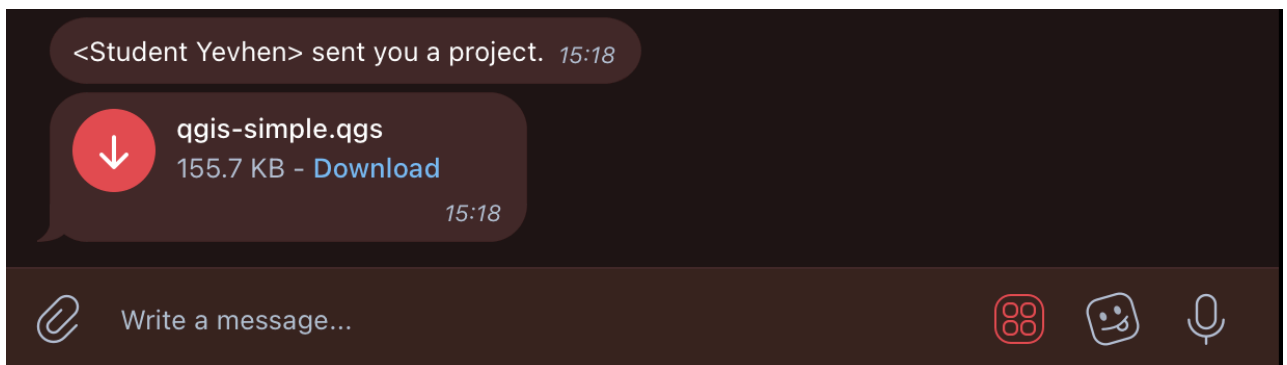


Рис. 3.15. Повідомлення викладача про здачу роботи студентом

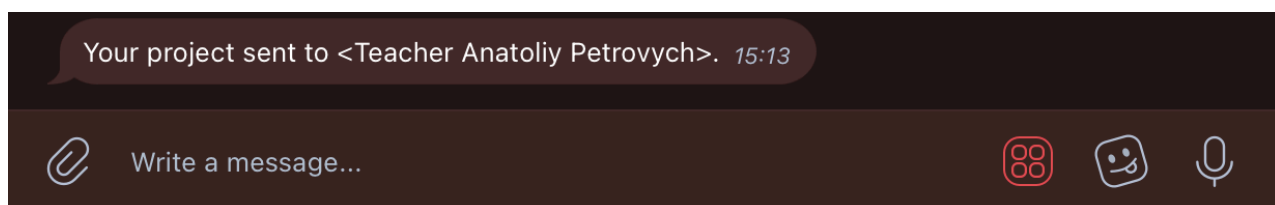


Рис. 3.16. Повідомлення студента про здачу роботи викладачеві

Ознайомитися із кодом даного програмного продукту можна переглянувши додаток «Б»

ВИСНОВКИ

1. Ефективне функціонування системи геодезії та досягнення її мети в умовах розвитку інформаційного суспільства неможливе без впровадження цифрових технологій. Саме тому актуальним завданням сьогодення є набуття відповідних компетентностей майбутніми фахівцями у цій галузі.
2. Аналіз літературних джерел щодо удосконалення і подальшого розвитку галузі свідчить про необхідність змін у програмах підготовки геодезистів доповненням існуючих навчальних дисциплін компонентами, які б забезпечили їх цифровими компетентностями високого рівня.
3. Потужним і доступним інструментом підвищення цифрової компетентності геодезиста є мова програмування Python. Це пояснюється рядом притаманних їй рис, таких, як легкість її вивчення, зрозумілий синтаксис, наявність численних бібліотек, що розширюють її стандартні можливості та широкий спектр використання.
4. Практичне опанування названої вище мови програмування значно розширило мою здатність оперувати в цифровому середовищі та спонукало до практичного її застосування у площині професійної підготовки – створення власного програмного продукту – плагіну для програмного продукту QGIS. Широкий спектр можливостей практичного застосування розробки та власне задоволення від роботи над ним дають підстави вважати, що використання сучасних мов програмування сприяє підвищенню творчої активності та набуття необхідних фахових компетентностей високого рівня.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Змішане» майбутнє: діджиталізація професійно-технічної освіти в Україні [Електронний ресурс] / Громадська організація “Інформаційно-аналітичний центр “Громадський Простір”. – 2021. – Режим доступу: <https://www.prostir.ua/?news=zmishane-majbutnje-didzhytalizatsiya-profesijno-tehnichnoji-osvity-v-ukrajini>.
2. Концепція розвитку цифрової економіки та суспільства України [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/67-2018-p#Text>.
3. Оновлення змісту професійної підготовки кваліфікованих робітників комерційного профілю [Електронний ресурс]. – Режим доступу до ресурсу: https://eu.docworkspace.com/d/sIIbPz9pp_m5hAY.
4. Географічна освіта України [Електронний ресурс]. - Режими доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%87%D0%BD%D0%B0_%D0%BE%D1%81%D0%B2%D1%96%D1%82%D0%B0_%D0%B2_%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%96
5. Професійно-технічна освіта [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Професійно-технічна_освіта.
6. В. Радкевич. Теоретичні та методичні засади розвитку професійної освіти і навчання. Результати, проблеми та перспективи / В. Радкевич. // Наук. вісник Інст-ту проф.-тех. освіти НАПН України. Професійна педагогіка. – 2016. - № 11. – С. 5-22. <https://doi.org/10.32835/2223-5752.2016.11.5-22>.
7. Гуревич Р.С. Інформатизація освіти – важливий чинник розвитку суспільства XXI століття / Р.С. Гуревич // Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. – 2016. – Вип. 47. – С. 5-10.
8. Міністерство цифрової трансформації. Концепція розвитку цифрових компетентностей [Електронний ресурс] / Міністерство цифрової трансформації.

<https://thedigital.gov.ua/news/kabmin-skhhvaliv-kontseptsuyu-rozvitku-tsifr>

9. QGIS. Ukrainian open initiative of creating tools for spatial planning in the QGIS environment [Електронний ресурс] / QGIS. – Режим доступу:

https://qgis.org/project/case-studies/ukrainian_spt/

10. Academic Determination of Teachical Information Optimization Due To Information and Communication Technologies / D. Kovalenko, N. Briukhanova, O. Kupriyanov, T. Kalinichenko // In: Auer M., Tsiatsos T. (eds). The Challenges of the Digital Transformation in Education. ICL 2018. Advances in Intelligent Systems and Computing, vol 917. – Springer, Cham., 2018. – P. 25-34.

11. Ковальчук В.І. Проблеми цифровізації фахової підготовки в закладах професійної освіти / В.І. Ковальчук // Матеріали II Міжнар. наук.-практ. конференції «Актуальні проблеми технологічної і професійної освіти» (Глухів, 14 травня 2020 р.). – Глухів: Глух. НПУ ім. О. Довженка, 2020. – С. 40-43.

12. Сидоренко В.В. Розвиток Soft Skills педагога професійної (професійно-технічної) освіти в умовах діджиталізації [Електронний ресурс] /

В.В. Сидоренко. – Режим доступу:

https://lib.iitta.gov.ua/722985/1/тези_Сидоренко_Глухів.pdf.

13. Ребенок В.М. Психолого-педагогічні основи використання інформаційних технологій викладачами професійної підготовки у закладах вищої освіти / В.М. Ребенок, О.М. Торубара // Вісник Нац. унів-ту «Чернігівський колегіум» ім. Т.Г. Шевченка. Серія: Педагогічні науки. – 2019. – Вип. 4 (160). – С. 35-40.

14. Alla Mutka K. Mapping Digital Competence: Towards a Conceptual Understanding. Monograph. Luxembourg, 2011. – P. 8-63. URL:

https://www.researchgate.net/publication/340375234_Mapping_Digital_Competence_Towards_a_Conceptual_Understanding.

15. Карташова Л.А. Розвиток цифрової компетентності педагога в інформаційноосвітньому середовищі закладу загальної середньої освіти / Л.А. Карташова, Н.В. Бахмат, І.В. Пліш // Інформаційні технології і засоби навчання.

– 2018. – Т. 68, № 6. – С. 193-205.

16. Трифонова О.М. Інформаційно-цифрова компетентність: зарубіжний та вітчизняний досвід / О.М. Трифонова // Наукові записки ЦДПУ ім. В. Винниченка. Серія: Педагогічні науки. 2018. – Вип. 173, ч. II. – С. 221-225.

17. Confronting the Challenges of Participatory Culture: Media Education for the 21st Century / Н. Jenkins, К. Clinton, R. Purushotma, A.J. Robinson, M. Weigel. – Chicago, Illinois : MacArthur Found., 2009. – 68 p.

18. Гаврилюк Л.Г. Цифрова культура, цифрова грамотність, цифрова компетентність як сучасні освітні феномени / Гаврилюк Л.Г., Топольник Я.В. // Інформаційні технології і засоби навчання. – 2017. – Т. 61, вип. 5. – С. 1-14.

19. DigComp 2.0: The Digital Competence Framework for Citizens. Update Phase 1: the Conceptual Reference Model. November 15, 2018. URL: <https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-researchreports/digcomp-20-digital-competence-framework-citizens-update-phase-1-conceptualreference-model>.

20. О. М. Васильєв. Програмування мовою Python / О. М. Васильєв., 2019. – 502 с.

21. Монті Пайтон [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Монті_Пайтон.

22. Копей В. Б. Мова програмування Python для інженерів і науковців [Електронний ресурс] / Копей В. Б. – Режим доступу до ресурсу: <https://vkokey.github.io/Python-for-engineers-and-scientists/#Найпростіша%20програма>.

23. Навіщо вивчати саме Python [Електронний ресурс] – Режим доступу до ресурсу: <https://dystosvita.org.ua/mod/page/view.php?id=663>.

24. Popularity of Programming Language [Електронний ресурс] – Режим доступу до ресурсу: <http://pypl.github.io/PYPL.html>

25. TIOBE Software BV[Електронний ресурс] – Режим доступу до ресурсу: <https://www.tiobe.com/tiobe-index/>

26. JetBrains s.r.o. Developed with drive and IntelliJ IDEA [Електронний ресурс] Режим доступу до ресурсу: <https://www.jetbrains.com/>
27. 5 способів вивчити Python [Електронний ресурс] – Режим доступу до ресурсу: <https://codeguida.com/post/715>.
28. Сидоренко Вікторія Вікторівна. Розвиток педагога професійної (професійно-технічної) освіти в умовах діджиталізації [Електронний ресурс] / Сидоренко Вікторія Вікторівна // Глухів. – 2020. – Режим доступу до ресурсу: https://lib.iitta.gov.ua/722985/1/тези_Сидоренко_Глухів.pdf.
29. Сидоренко Вікторія. Професійний розвиток фахівців в умовах формальної, неформальної та інформальної освіти: ключові компетентності і ресурси : електронний курс. Київ: ДУ «НМЦ «Агроосвіта», 2019. 130 с.
30. Цифрова компетентність вчителя [Електронний ресурс] // Київ. – 2019. – Режим доступу до ресурсу: <https://lib.iitta.gov.ua/715564/1/Цифрова%20компетентність%20сучасного%20вчителя%20нової%20української%20школи%202019.pdf>
31. Python став найпопулярнішою мовою на GitHub [Електронний ресурс]. – 2024. Режим доступу до ресурсу: <https://dou.ua/forums/topic/50957/>
32. Вдовичин Т. Я., Лазурчак Л. В. Навчання основ програмування студентів фізико-математичного профілю. // Інформаційні технології в освіті. — 2017, №2
33. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>.
34. Welcome to aiogram's documentation! [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aiogram.dev/en/latest/>.
35. PostgreSQL - Офіційна документація [Електронний ресурс] / PostgreSQL– Режим доступу до ресурсу: <https://www.postgresql.org/about/>
36. Розуміння Docker [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/post/253877/>.
37. PyMongo [Електронний ресурс] – Режим доступу до ресурсу: <https://pymongo.readthedocs.io/>

38. Etern Noir. Документація PytelegramBotAPI [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/pyTelegramBotAPI/>
39. QGIS. Документація PyQGIS[Електронний ресурс] – Режим доступу до ресурсу: <https://qgis.org/pyqgis/3.16/core/QgsApplication.html>
40. Tiangolo. Документація FastAPI [Електронний ресурс] – Режим доступу до ресурсу: <https://fastapi.tiangolo.com/>
41. Mongo team. Документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>
42. Діаграма прецедентів [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Діаграма_прецедентів

ДОДАТОК А. СЕРТИФІКАТИ ПРО ЗАКІНЧЕННЯ КУРСІВ ІЗ МОВИ ПРОГРАМУВАННЯ PYTHON



Сертифікат про закінчення курсу з основ Python на платформі Coursera від Мічиганського університету

PCAP: Programming Essentials in Python

During the Cisco Networking Academy® course, administered by the undersigned instructor, the student has studied the following skills:

- the universal concepts of computer programming (i.e. variables, flow control, data structures, algorithms, conditional execution, loops, functions, etc.)
- developer tools, developer tools and the runtime environment;
- the syntax and semantics of the Python language;
- the fundamentals of object-oriented programming and the way they are adopted in Python;
- the means by which to resolve typical implementation problems;
- the writing of Python programs using standard language infrastructure;
- fundamental programming techniques, best practices, customs and vocabulary, including the most common library functions in Python 3.

This Statement of Achievement acknowledges that during the course PCAP: Programming Essentials in Python, the student has been able to accomplish coding tasks related to the basics of programming, and understands the programming techniques, customs and vocabulary used in the Python language.

By completing the course, the student is now ready to attempt the qualification PCAP – Certified Associate in Python Programming certification, from the OpenEDG Python Institute.

Taras Shyichuk

Student

Чернівецький національний університет

Academy Name

Ukraine

Location

4 Dec 2020

Date

Yuliya Tanasyuk

Instructor

Instructor Signature

www.netacad.com | www.pythoninstitute.org

Сертифікат про закінчення курсу по Python від CISCO



Сертифікат про закінчення курсу «Основи Python» від компанії Datawiz

softserve | IT academy

Series CN № 5808/2021

Click to verify



CERTIFICATE

Is issued to certify that

Taras Shyichuk

has successfully completed SoftServe IT Academy course

**DEVELOPMENT, DESIGN, AND ARCHITECTURE OF MODERN SOFTWARE SYSTEMS BASED ON
OBJECT-ORIENTED LANGUAGES. PYTHON.**Chris Baker
CEO, SoftServe Inc.Chernivtsi, Ukraine
June 17, 2021

Сертифікат про закінчення курсу «Розробка, дизайн та архітектура сучасних програмних систем орієнтованих на об'єктно-орієнтованих мовах. Python» від компанії SoftServe

ДОДАТОК Б. КОД ПРОГРАМНОГО ПРОДУКТУ**/models.py**

```
from dataclasses import dataclass, asdict, fields
from enum import Enum
from typing import Optional, Union

import pymongo

mongo_client = pymongo.MongoClient("mongodb://localhost:27017")
user_collection = mongo_client['admin']['user']

class MongoMixin:
    @classmethod
    def get(cls, query: dict) -> Optional["UserModel"]:
        doc = user_collection.find_one(query)
        return doc and cls.from_dict(doc) # noqa

    @classmethod
    def list(cls, query: dict) -> list["UserModel"]:
        return [cls.from_dict(user) for user in user_collection.find(query)] # noqa

    def save(self) -> "UserModel":
        user_collection.insert_one(self.to_dict()) # noqa
        return self # noqa

    def update(self, data: dict) -> "UserModel":
        res = user_collection.update_one({"chat_id": self.chat_id}, {"$set": data})
```

```
# noqa
```

```
    return self.get({"chat_id": self.chat_id}) # noqa
```

```
class UserType(Enum):
```

```
    TEACHER = 'teacher'
```

```
    STUDENT = 'student'
```

```
@dataclass
```

```
class UserModel(MongoMixin):
```

```
    chat_id: int
```

```
    first_name: str
```

```
    last_name: Optional[str]
```

```
    phone: str
```

```
    type: Union[UserType, str] = UserType.STUDENT.value
```

```
@classmethod
```

```
    def from_dict(cls, data: dict) -> "UserModel":
```

```
        return cls(**{field.name: data.get(field.name) for field in fields(cls)})
```

```
    def to_dict(self) -> dict:
```

```
        return asdict(self) # noqa
```

```
    def __str__(self):
```

```
        return f"<{self.type.capitalize()} {self.first_name} {self.last_name or  
\"}\".strip() + ">"
```

/bot.py

```
import telebot
from telebot import types
from telebot.states import State, StatesGroup
from telebot.storage import StateMemoryStorage

from models import UserModel, UserType

state_storage = StateMemoryStorage()
bot = telebot.TeleBot('7807410893:AAH3ewlsO0H8sGKqulmrk2CZ0lFZH3csip8',
state_storage=state_storage)
```

```
class UserStates(StatesGroup):
    approve_password = State()
```

```
@bot.message_handler(commands=['start'])
def start(message: types.Message):
    bot.reply_to(
        message,
        "Welcome to the QGIS Telegram Bot!\n"
        "You can get your chat id by running /register command."
    )
```

```
@bot.message_handler(commands=['register'])
```

```

def chat(message: types.Message):
    keyboard = types.ReplyKeyboardMarkup(one_time_keyboard=True)
    keyboard.add(types.KeyboardButton(text="Share your contact for
registration", request_contact=True))
    bot.send_message(
        message.chat.id,
        "You should share your phone number",
        reply_markup=keyboard
    )

```

```
@bot.message_handler(content_types=['contact'])
```

```

def contact(message: types.Message):
    if UserModel.get({'chat_id': message.chat.id}):
        bot.reply_to(message, "You already registered.")
    return

```

```

user = UserModel(
    chat_id=message.chat.id,
    first_name=message.contact.first_name,
    last_name=message.contact.last_name,
    phone=message.contact.phone_number
)
user.save()
bot.reply_to(
    message,
    f"[{user.chat_id}] {user.first_name}, "
    f"you registered successfully as {user.type}.",
)

```



```

@bot.message_handler(commands=['as_teacher'])
def as_teacher(message: types.Message, ):
    if not (user := UserModel.get({'chat_id': message.chat.id})):
        bot.reply_to(message, "You should register first.")
        return

    bot.reply_to(message, "Please provide the secret password to continue.")
    bot.set_state(message.from_user.id, str(UserStates.approve_password),
message.chat.id)

    with bot.retrieve_data(message.from_user.id, message.chat.id) as data:
        data['user'] = user

@bot.message_handler(
    func=lambda message: bot.get_state(message.from_user.id, message.chat.id)
== str(UserStates.approve_password))
def approve_password(message: types.Message):
    with bot.retrieve_data(message.from_user.id, message.chat.id) as data:
        user = data['user']

    if message.text != "super secret":
        bot.reply_to(message, "Invalid password. You are not authorized to become
a teacher.")
        return

    user.update({'type': UserType.TEACHER.value})
    bot.reply_to(message, f"{user.first_name}, you are
{UserType.TEACHER.value} now.")
    bot.delete_state(message.from_user.id, message.chat.id)

```

```

@bot.message_handler(commands=['students'])
def students(message: types.Message):
    if not (user := UserModel.get({'chat_id': message.chat.id})):
        bot.reply_to(message, "You should register first.")
        return

    if user.type != UserType.TEACHER.value:
        bot.reply_to(message, "You are not authorized to view the list of students.")
        return

    if not (student_users := UserModel.list({'type': UserType.STUDENT.value})):
        bot.reply_to(message, "No students registered yet.")
        return

    bot.reply_to(message, "Students:")
    bot.send_message(
        message.chat.id,
        "".join([
            f"\n[ {student.chat_id}] {student.first_name}"
            for student in student_users
        ])
    )

if __name__ == '__main__':
    bot.infinity_polling()

```

/plugin.py

```

from qgis._core import Qgis
from qgis.core import QgsProject
from qgis.utils import iface
from typing import Optional

import telebot
from PyQt5.QtCore import QObject

from models import UserType, UserModel

bot =
telebot.TeleBot('7807410893:AAH3ewlsO0H8sGKqulmrk2CZ0lFZH3csip8')

```

```

class TelegramGis(QObject):
    def __init__(self):
        super().__init__()
        self.qgs_project = QgsProject.instance()
        self.iface = iface
        self.register()

        self._teacher = None
        self._student = None

    @property
    def teacher(self) -> Optional[UserModel]:
        if self._teacher is None:

```

```
self._teacher = UserModel.get({'type': UserType.TEACHER.value})
```

```
if self._teacher is None:
```

```
    iface.messageBar().pushMessage(
```

```
        "Teacher isn't registered.",
```

```
        (
```

```
            "Notify the teacher to register."
```

```
            "Use command /register and /as_teacher to set `teacher` status."
```

```
        ),
```

```
        level=Qgis.Critical
```

```
    )
```

```
    return
```

```
return self._teacher
```

```
@property
```

```
def student(self) -> Optional[UserModel]:
```

```
    if self._student is None:
```

```
        iface.messageBar().pushMessage(
```

```
            "Student isn't registered.",
```

```
            (
```

```
                "Please use command /register to register. "
```

```
                "Then on QGIS Python console provide `telegram_gis.student =
```

```
your_chat_id`."
```

```
            ),
```

```
            level=Qgis.Critical
```

```
        )
```

```
    return self._student
```

```
@student.setter
```

```

def student(self, value: int):
    self._student = UserModel.get({'chat_id': value})

def register(self):
    self.qgs_project.projectSaved.connect(self.on_project_save)

def on_project_save(self):
    if self.teacher is None:
        return

    if self.student is None:
        return

    doc = open(self.qgs_project.fileName(), 'rb')
    bot.send_message(
        self.teacher.chat_id,
        f'{self.student} sent you a project.',
    )
    bot.send_document(self.teacher.chat_id, doc)
    bot.send_message(self.student.chat_id, f"Your project sent to
{self.teacher}.")
    doc.close()

telegram_gis = TelegramGis()

```