

**Саган Б.Г.**

*студент,*

*Науковий керівник: Ковалюк Т.В.*

*доцент,*

*Національний технічний університет України*

*«Київський політехнічний інститут імені Ігоря Сікорського»*

## **ЗАДАЧА ПОБУДОВИ ІНДИВІДУАЛЬНОГО ПЛАНУ НАВЧАННЯ СТУДЕНТА НА ОСНОВІ ЙОГО КОМПЕТЕНТНОСТЕЙ ТА ПОБАЖАНЬ**

В статті пропонується розглянути новий підхід, щодо побудови індивідуального навчального плану студента. Даний підхід заснований на врахуванні актуальних вимог ринку щодо випускників певної спеціальності і профілю та особистих побажаннях студента у вивченні тих чи інших дисциплін.

Як ми всі знаємо одним з найважливіших завдань вищих навчальних закладів є задоволення ринку праці в спеціалістах, яких він потребує. При цьому випускники повинні мати високий рівень знань в галузі у якій вони будуть працювати. Також слід зауважити, що проблема актуальності знань які надаються студентам при їх навчанні в ВНЗ є особливо критичною в сфері ІТ. На сьогоднішній день ІТ ринок розвивається надзвичайно стрімко і технології які вивчаються студентом на перших курсах можуть бути не потрібними на момент закінчення навчання.

Ще одним важливим фактором є визначення конкретної області розвитку студента, в якій він планує працювати після закінчення ВНЗ. Для прикладу ми можемо розглянути дві галузі ІТ: e-Commerce та VR (Virtual Reality). Вимоги до спеціалістів даних галузей абсолютно різні. Якщо студент хоче працювати в електронній комерції йому навряд чи знадобляться знання низько-рівневих мов програмування. Тобто, нам потрібно надавати студенту набір тих дисциплін які є необхідними для роботи в обраній ним галузі та мінімізувати вивчення дисциплін які не будуть використовуватись в майбутньому і не є фундаментальними.

Також необхідним є можливість зміни навчального плану студента. Зміна може бути ініційована стейк-холдерами, наприклад, у разі виникнення потреби в знанні додаткових технологій або заміни початкового стеку технологій, чи бажанням студента змінити напрямок свого розвитку. Тобто, якщо студент після одного року вивчення певної галузі захоче її змінити, то у нього буде можливість побудови нового навчального плану на основі тих дисциплін які будуть запропоновані в новій галузі, а також тих які він уже освоїв за перший рік навчання, якщо вони пов'язані з новим напрямком розвитку.

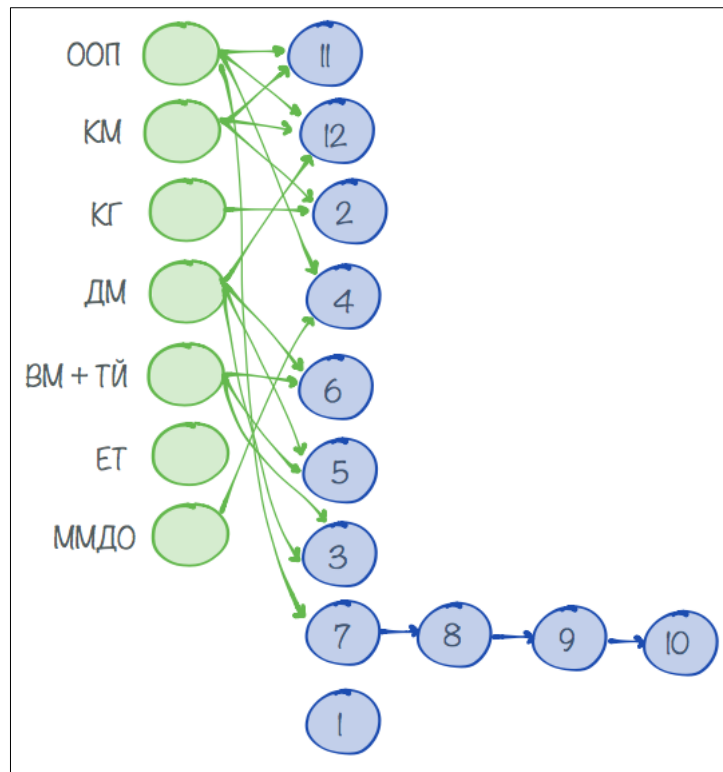
Давайте детальніше розглянемо процес побудови навчального плану.

*Першим етапом* побудови навчального плану є побудова графу передування дисциплін (Рис. 1).

*Граф передування дисциплін* – це сукупність об'єктів(дисциплін) та зв'язків(можливості вивчати певну дисципліну після освоєння попередньої) між

ними. Дисципліни будуть розглядатись як вершини, або вузли графу, а зв'язки – як орієнтовані ребра. Це означає що, в нашій задачі будову дисциплін можна змоделювати за допомогою орієнтованого мультиграфу, в якому вершини – це дисципліни, а дуги (орієнтовані ребра) – посилення на інші дициспліни, які можна освоїти.

В подальшому даний граф буде використуватась для перевірки чи досягне студент повного опанування обраного профелю. Тобто, всі компетенції обраного профелю будуть освоєні.



**Рис. 1. Граф передування дисциплін**

*Другим етапом* є визначення пріоритетів дисциплінам. Студент каже які дисципліни він хоче вивчати понад усе.

*Третім етапом* додавання до плану дисциплін які необхідні для освоєння дисциплін, що обрав студент, у разі якщо його компетентностей не вистачає, та тих які допоможуть повністю опанувати всі необхідні компетенції обраного профелю.

*Четвертим етапом* є визначення кількості кредитів для кожної дисципліни яка ввійшла в навчальний план студента. Кількість кредитів буде розрахуватись в залежності від важливості компетентностей які покриває дисципліна та побажань студента.

Таким чином, ми отримаємо індивідуальний навчальний план студента, що задовільняє його побажання та відповідає актуальним вимогам ринку. Для підвищення якості підготовки студентів даний процес буде проводитись кожного семестру на основі оновлених вимог ринку. Слід зауважити, що в кінці семестру студент проходить певну атестацію, після якої ми будемо знати його

поточний рівень і на наступній ітерації ми будемо використовувати ці дані для виключення з плану дисциплін, що дублюються.

Основною метою даної статті є демонстрацію нового підходу до побудови індивідуального навчального плану студента на основі його компетентностей та побажань. Так як, методи побудови навчальних планів, що використовуються зараз є дещо застарілими і не відповідають теперішнім вимогам.

### **Список використаних джерел:**

1. Кривицька М.А., Бушмелева К.И. Підходи до проектування задачі складання робочого навчального плану спеціальності вищої професійної освіти. [Текст]
2. Тім Лорман, Джоан Депплер, Девід Харві. Інклюзивний підхід до побудови навчального процесу. [Текст]
3. Кристофидес Н. Теория графов. Алгоритмический подход. Москва, 1978.

**Скороходов К.Ф.**

*студент,*

*Харківський національний університет радіоелектроніки*

## **ЗАСТОСУВАННЯ МЕТОДУ ВІДБИТКІВ ДЛЯ ШВИДКОГО ПОШУКУ ПЛАГІАТУ В КОДІ ПРОГРАМИ**

Цифрові документи, в тому числі і програмний код, дуже легко скопіювати. На жаль, деякі люди користуються цим для того, щоб видавати чужі роботи за свої. Проблема плагіату є дуже важливою, наприклад, в галузі освіти, коли студенти, замість того, щоб виконувати завдання самостійно, копіюють роботи інших студентів, можливо, лише вносячи при цьому незначні зміни. Вручну перевіряти на плагіат велику кількість робіт дуже складно, тому, в такому випадку, викладачеві було б зручно мати інструмент, який би дозволив автоматично перевірити роботи та знайти серед них підозріло схожі один до одного. Таке програмне забезпечення може приймати на вхід вихідні коди, аналізувати їх, і видавати ранжирований список пар схожих файлів, а також якимось чином виділяти схожі рядки, щоб користувач мав можливість оцінити, чи дійсно плагіат має місце в цьому випадку.

На сьогодні було розроблено багато різноманітних підходів для пошуку плагіату в вихідному коді програм. Одним із підходів є підрахунок певних атрибутів програмного коду, наприклад кількості унікальних операторів та операндів мови програмування, якою написана програма, загальної кількості вживань певних операторів та операндів та ін. Іншим підходом є складання дерев розбору програм та порівняння їх за допомогою складних алгоритмів [1].

Одними з найбільш ефективних алгоритмів для швидкого пошуку однакових частин програмного коду у багатьох документах є алгоритми на основі відбитків. Ми розглянемо так званий алгоритм провіювання (англ. winnowing), який відноситься до цієї категорії [2].