

Активне навчання

практика та виклики для дисциплін технічного та природничого циклу

Олексій Молчановський, старший викладач ФПН, керівник магістерської освітньої програми з комп'ютерних наук УКУ

Львів, Український католицький університет

13 лютого 2018 р.

Питання семінару

1. Мотивація до використання методів активного навчання.
Короткий підсумок матеріалів UNESCO під загальною назвою THE FUTURES of LEARNING (2015 рік).
2. Особистий досвід впровадження активного навчання на курсі «Алгоритми та структури даних» для студентів бакалаврської програми з комп'ютерних наук Українського Католицького Університету.
3. Розгляд кейсів та обговорення можливостей впровадження методів активного навчання на інших курсах.

“It’s not what you give them,
it’s what they take away counts.”

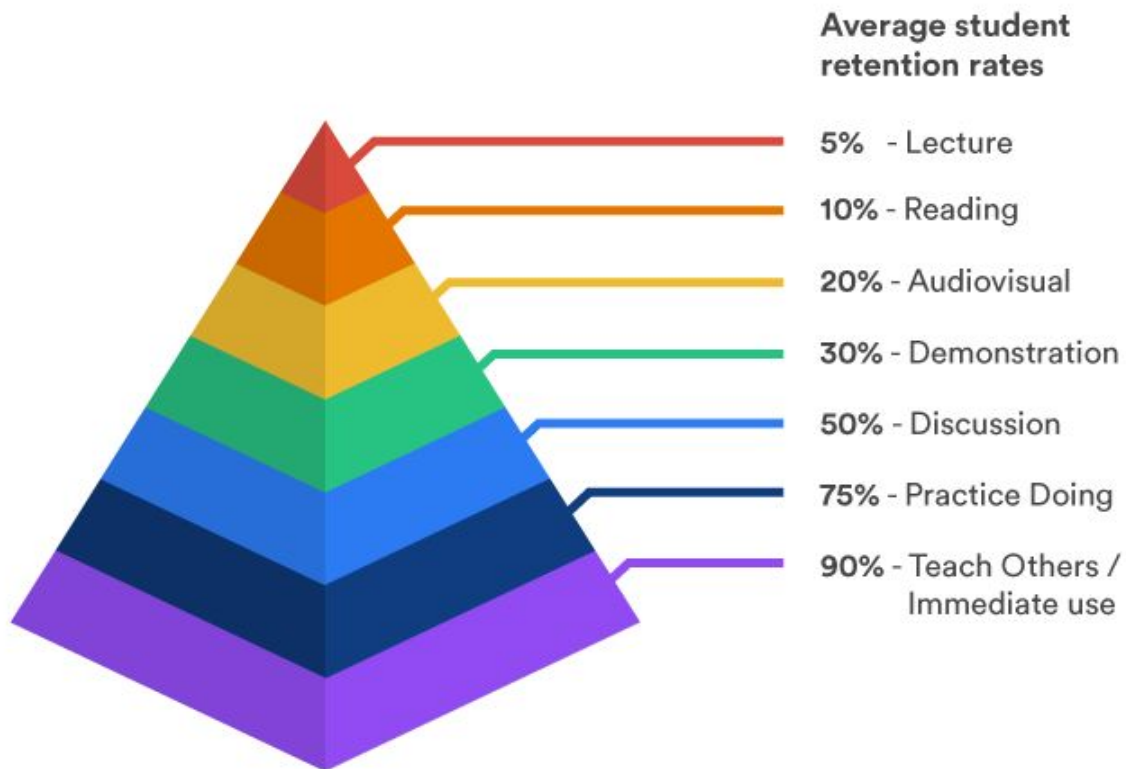
*“Active Training: A Handbook of Techniques,
Designs, Case Examples, and Tips”
Melvin L. Silberman, Carol Auerbach*

The Future of Learning, UNESCO, 2015

Above all, studies have found that learners are more successful at acquiring new competencies when they build strong metacognitive abilities, reflect objectively on new concepts learned, and integrate that information with their existing knowledge and skills.

The process of adapting new knowledge for their own use and incorporating it into their existing conceptual frameworks will support further learning. Once new learning is integrated into existing 'ways of knowing', this in turn nurtures creativity and originality and establishes new cognitive habits.

Learning Pyramid



«Алгоритми та структури даних»

Вхідні умови

- Студенти другого року навчання
- Види занять в класі:
 - лекції раз на тиждень
 - практичні заняття раз на два тижні
 - консультації
- Види самостійної роботи:
 - виконання практичних завдань (програмування алгоритмів)
- Наявні матеріали:
 - Конспект лекцій, підручник англійською мовою, онлайн курс на Prometheus, додаткові відео-матеріали

Проблеми минулого року

- Студенти НЕ читають підручник та інші довгі тексти
- Студенти НЕ готуються наперед до семінарів
 - не дивляться задачі, які анонсовані заздалегідь до семінару
- Студенти НЕ вміють писати та читати псевдокод

Активне навчання на курсі

- Основний принцип: залучати студентів до активного пропрацювання навчального матеріалу
- Типи активностей:
 - лекції
 - семінарські заняття

Вступна лекція

- Мотивація:
 - познайомитися зі студентами
 - розповісти про курс
 - зрозуміти мотивацію студентів
 - задати правильні очікування
 - встановити діалог

Вступна лекція: мотивація

Навіщо вивчати алгоритми?

● «ЗА»

- Покращити швидкість думки
- Щоб навчитись робити захищений бізнес
- Щоб мати готову відповідь на вирішення конкретної проблеми
- Зрозуміти роботу з великими даними
- Як програміст, якщо ти знаєш алгоритми, то ти цінуєшся більше
- Щоб заробляти більше грошей
- Щоб пройти співбесіду
- Дисциплінованість та планованість своєї роботи
- Ефективність коду
- Можливість розв'язувати швидше основну задачу

● «ПРОТИ»

- Я роблю сайти, навіщо мені алгоритми
- Це не моє-твоє
- Якщо ти вже пройшов цей курс
- Цей предмет стандартизує розум
- Немає нічого кращого ніж бінарний пошук, а я його вже знаю!!!
- Ми не вчимо самі алгоритми, а це демотивує
- Частина алгоритмів це загальна логіка
- Вивчення алгоритмів не робить тебе кращим програмістом
- Не використовується "на галерах" :(
- Більшість алгоритмів вже реалізовані в різних бібліотеках

Вступна лекція: ad fontes (до глибин)

Ad fontes

- *Вивчення математичної моделі допомагає розумінню алгоритму. Більшість алгоритмів реалізовані, але ти не знаєш як вони зроблені і ти не можеш ефективно його використовувати*
- *Зрозуміти еволюцію алгоритмів*
- *Починати від простого і йти до складніших алгоритмів*
- *Зрозуміти причини творення конкретних алгоритмів*
- *Важливо щоб вміти добре пояснити іншій людині*
- *Зрозумівши попередній досвід творення алгоритмів, ми можемо спроектувати на нову систему та для нових потреб*
- *Якщо ти знаєш, як конкретно вирішуються старі проблеми, то можеш не повторюватися, а вирішувати нові*

Вступна лекція: очікування від курсу

Очікування від курсу

- *Курс має бути цікавим (як мінімум)*
- *Це буде не найпростіший курс*
- *Пройдемо реалізацію деяких алгоритмів*
- *Робота в групах, дається конкретна проблема і треба розробити найкращий алгоритм і довести його ефективність*
- *На виході курсу я буду знати “на зубок” алгоритми, бачити патерни*
- *Багато алгоритмів буде візуалізовано*
- *В кінці курсу ми будемо реально використовувати ці алгоритми*

Вступна лекція: очікувані цілі (від викладача)

Очікувані цілі на курсі

- **Навчитися:**
 - Писати псевдокод алгоритмів
 - На основі псевдокоду писати власний код
 - Аналізувати алгоритм/код для визначення складності
 - Обґрунтовувати/доводити коректність алгоритмів
 - Створювати власні структури даних
 - Реалізовувати різні стратегії: декомпозиція, жадібна, динамічне програмування, ...
- **Розуміти:**
 - Класи складностей задач
 - Обмеження застосування алгоритмів для різних класів

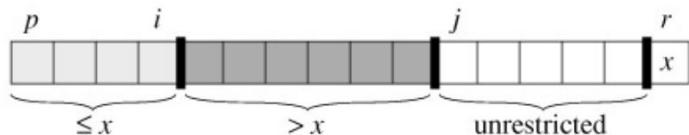
Активності на лекціях

- Мета:
 - дати студентам елемент самостійного зворотного зв'язку (feedback loop), щоби вони «зрозуміли, що саме вони щойно зрозуміли»
- Метод:
 - короткі завдання (5 хв) на тему, яка щойно була розглянута
 - дві-три подібні сесії протягом лекції
 - можливість працювати у парі з сусідом
 - завдання можуть бути однакові для всіх чи різні, кооперативні (робимо разом) або конкурентні (завадити один одному)

Активності на лекціях (приклад 1)

Швидке сортування. Розбиття масиву. Приклад

В процесі роботи процедури Partition масив $A[p \dots r]$ складається з чотирьох частин:



- якщо $p \leq k \leq i$, то $A[k] \leq x$
- якщо $i + 1 \leq k \leq j - 1$, то $A[k] > x$
- якщо $k = r$, то $A[k] = x$
- якщо $j \leq k \leq r - 1$, то елементи можуть мати довільні значення

Завдання

1. Запишіть псевдокод процедури $\text{Partition}(A, p, r)$
2. Перевірте на прикладі $[2, 8, 7, 1, 3, 5, 6, 4]$ або на будь-якому іншому прикладі
3. Визначте час роботи процедури Partition

Активності на лекціях (приклад 2)

Інверсії

- $A[1\dots n]$ – це масив, який складається з n різних чисел
- Якщо $i < j$ та $A[i] > A[j]$, то пара (i, j) називається інверсією в масиві A

Завдання

- Скільки інверсій існує в масиві $A = [5, 2, 4, 3, 1]$?
- Яка найбільша можлива кількість інверсій для масиву розмірності n ?

Активності на лекціях (приклад 3)

Знайдіть контрприклад

- Ви повинні обрати один з чотирьох критеріїв. Для критерію, обраного вашим сусідом, ви повинні знайти контрприклад, який спростовує коректність жадібного підходу
 - A. за зростанням часу s_i (найбільш ранній час початку)
 - B. за зростанням часу f_i (найбільш ранній час закінчення)
 - C. за збільшенням тривалості процесу $f_i - s_i$ (найкоротший інтервал)
 - D. для кожної задачі i підрахувати кількість несумісних процесів c_i ; впорядкувати за зростанням числа c_i (найменше конфліктів)

Активності на лекціях

Переваги

- Студенти починають акцентувати увагу на цих невеликих елементах самостійної роботи
- Абсолютна більшість студентів не намагається відлинути від виконання цих завдань
- Можна пройтися класом та дати короткі коментарі студентам
- Залучення всіх, а не окремих (як це буває, коли просто ставиш запитання в аудиторію) студентів до «активного думання»

Недоліки

- Забирає багато часу: виконання подібних завдань може займати до третини часу лекції
- Завдання треба добре проектувати заздалегідь: зрозумілість для студента, як студенти будуть комунікувати під час виконання завдання, ...

Щохвилининний план лекції

Лекція 1. Вступ до курсу.

Частина 1. Вступ.

1. **(2 хв)** Представлення себе
2. **Ice breaker.** Навести три твердження про те, що робив цього літа: два правдивих і одне хибне. Аудиторія має відгадати яке хибне.
3. **(2 хв)** Показати відео про тренування лижника
4. **(7 хв)** Чому важливо вивчати алгоритми?
 - (3 хв) Попарне обговорення.
 - (3 хв) Окремі студенти оголошують своє бачення вголос. Записати на слайд
 - Решта відповідей може бути кудись записана онлайн (Google doc, socrative, moodle)
5. **(5 хв)** Принцип "ad fontes" на курсі
 - Як ви розумієте принцип ad fontes на курсі алгоритмів?
 - (5 хв) Попарне обговорення. З оголошенням свого бачення.
6. **(7 хв)** Ваші очікування від курсу? Де ви плануєте використовувати набуті тут знання?
7. **(10 хв)** Попередження про експериментальний підхід на курсі.
 - Active training approach: active training occurs when the participants do most of the work. (взяти вступ з АТ, page 9-10)
 - Слайд про gap between knowing and doing
 - наголосити на важливості вмінь: писати псевдокод, доводити теореми, обґрунтовувати коректність алгоритмів
 - розповісти про проблеми минулого року на курсі та закликати спільно їх попереджати
 - піраміда сприйняття / отримання інформації (слухання - навчання)
8. **(3 хв)** Політики на курсі:
 - Оцінювання
 - Академічна доброчесність
9. **(10 хв)** Гість: Остап Столярчук
10. **(15 хв)** Провести опитування по знанню алгоритмів (див. опитувальник з попереднього року)

Завершення лекції

- Мета:
 - дати можливість студентам швидко оглянути всю лекцію: що насправді відбулося протягом останньої години?
- Метод:
 - опитати сусіда
 - заповнити гугл-форму чи форму на Мудлі
 - у випадку форми можливі додаткові бали за змістовні відповіді
- Бонус для викладача:
 - краще розумієш, на що саме студенти звернули увагу на лекції
 - незрозумілі елементи лекції можна розглянути на консультації

Завершення лекції

Запитайте в сусіда та дайте відповідь на питання

1. Що ви робили на цій лекції?
2. Що ви отримали після цього досвіду?
3. Чому ви навчилися за цю лекцію?

Підсумки лекції 06 Хеш таблиці

*Обов'язкове поле

Що ви зрозуміли з лекції *

Ваша відповідь

Що лишилося не зрозумілим *

Ваша відповідь

Ім'я, прізвище *

Ваша відповідь

НАДІСЛАТИ

Ніколи не вказуйте паролі в Google Формам.

Додаткові активності на лекціях

- Перегляд відео-фрагментів, які пояснюють деякі елементи лекції або демонструють приклади
 - Зокрема, це дозволяє перемкнути увагу студентів
- Запрошення гостей (через скайп)
 - Основна проблема з гостями: проговорити, про що вони мають спілкуватися зі студентами

Семінарські заняття

- Мотивація:
 - Підхід «розв'язання задач біля дошки» викладачем або окремим студентом вже не працює (принаймні, в мене)
 - Чому? – Більшість студентів швидко втрачає концентрацію
- Мета:
 - Спонування студентів до попереднього ознайомлення з задачами (а-ля домашні завдання)
 - Робота в командах
 - Залучення найбільш дієвого способу навчання – «навчання інших»

Семінарські заняття: метод

- Студенти працюють в групах (до 6 осіб)
- Завдання видаються заздалегідь (за кілька днів)
- Завдання складаються з кількох задач
- Задача захищається одним або кількома членами команди
- Здана задача зараховується одразу всім членам команди
 - Окрім найбільш кричущих випадків, коли хтось явно нічого не робить
- Для швидкого обліку зданих задач можна використовувати прості картки



(ім'я, прізвище)		
1	2	3
4	5	6

Семінарські заняття: слабкі місця підходу

- Як диференціювати оцінки всередині команди:
 - Як виявляти тих, хто більше привніс для розв'язання задачі?
 - Як виявляти тих, хто нічого не робить в команді?
 - Як комунікувати студентам ці різні бали?
- Як спонукати студентів рівномірно розподіляти роботу в команді?
 - Насправді, студенти досить швидко еволюціонують до цього
- Основна проблема – це час:
 - Необхідно слідкувати, щоби всім командам приділити достатньо часу
- Що робити з тими задачами, які не встигли розв'язати за заняття?

Семінарські заняття: що працює

- Студенти досить швидко привчаються до методу (головне не змінювати правила надто радикально від заняття до заняття)
 - Наприклад, студенти швидко самі починають попередньо розставляти столи для більш зручного розташування команд
- Коли в команді є один-два найрозумніших, які хочуть постійно здавати розв'язки, то спонукати їх пояснювати розв'язки іншим членам команди
- Команди рухаються в різному темпі і хтось може здати більше задач за інших – це нормально
- **Коли студенти пояснюють один одному задачі та знаходять помилки один в одного, тоді справді «формується нові нейронні зв'язки»**

Проміжне опитування студентів на курсі

Проміжне опитування щодо якості курсу «Алгоритми та структури даних»

1. Вкажіть, будь ласка, що Вам подобається в цьому курсі (організація, підходи, завдання, ...)?

2. Вкажіть, будь ласка, що Вам НЕ подобається в цьому курсі (організація, підходи, завдання, ...)?

3. Що би Ви змінили в курсі (проведення занять, організація, комунікація, завдання, ...)?

4. Що у цьому курсі у Вас викликає найбільші труднощі?

5. Оцініть наскільки протягом курсу Ви особисто розвинули наступні навички. (Поставте навпроти кожного пункту бал від 1 до 10, де 1 відповідає відсутності прогресу з набуття вказаної навички, 10 - максимальний прогрес.)

A) Писати псевдокод: _____

B) На основі псевдокоду писати реальний код: _____

C) Аналізувати алгоритми на складність: _____

D) Обґрунтовувати/доводити коректність алгоритмів: _____

6. Скільки годин протягом тижня Ви приділяєте в середньому на виконання завдань та самостійне опанування матеріалів курсу: _____

7. Якщо бажаєте отримати зворотний зв'язок від викладача, то можете лишити своє ім'я та прізвище:

Підсумки.
Запрошення до співпраці.
Що далі?