

Перевіряємо розуміння, а не звіт

Етап: концепція, підготовка пілоту на одному курсі · 10 жовтня 2026 · Чому.Lab — робоча назва

Чому.Lab — нова модель практичних завдань для computer science дисциплін. Студент обирає реальну задачу й перетворює її на математичну модель та інтерактивний артефакт будь-якими інструментами, зокрема ШІ. Потім коротко захищає роботу в аудиторії (online-зустрічі) перед ШІ-екзаменатором. Остаточну оцінку ставить викладач.

1. Проблема

Типова лабораторна робота з computer science виглядає так. Усі студенти отримують однакове завдання з методички за варіантом, розв'язують його і здають звіт за шаблоном. Оцінюють здебільшого оформлення та відповідність формальним критеріям: структуру звіту, наявність скриншотів, висновки за зразком.

Така практика погано перевіряє те, що потрібно інженеру:

- **розуміння**, чому рішення працює і коли перестане працювати;
- **прийняття рішень** там, де немає однієї правильної відповіді;
- **моделювання** — уміння перетворити «брудну» інформацію про реальний світ на математичну модель;
- **перевірюваний результат** — уміння зробити з моделі артефакт, який можна запустити, змінити й перевірити.

ШІ загострив проблему. Звіт за шаблоном тепер генерується за хвилину. Викладач отримує охайну роботу й не знає, чи студент її розуміє. Студенти природно обирають найлегший шлях, а нецікаві завдання не дають причини обирати інший.

2. Для кого

- **Викладачі** computer science дисциплін — користувачі. Отримують спосіб перевірити розуміння кожного студента без усного захисту кожної роботи власними силами.
- **Студенти** — користувачі. Обирають задачі, які їм цікаві, і працюють звичними інструментами, зокрема ШІ.
- **Кафедри й університети** — замовники. Отримують доказовий результат навчання, який можна показати під час акредитації.
- **Партнери** — компанії, лабораторії, громади. Дають реальні задачі й дані, а натомість отримують перші варіанти рішень і знайомство з сильними студентами.

3. Поточна практика і наша альтернатива

Поточна практика оптимізована під перевірку формальних ознак роботи. Ми пропонуємо оптимізувати її під перевірку розуміння.

	Типова лабораторна	Чому.Lab
Завдання	Однакове для всіх, з методички за варіантом	Обране студентом з банку реальних задач різної складності
Що оцінюють	Оформлення та формальні критерії	Розуміння, рішення, якість моделі
Результат	Статичний звіт	Інтерактивний артефакт, який можна запустити й змінити
Роль ШІ	Формально заборонений, фактично використовується потай	Дозволений під час виконання, використання видно
Перевірка	Читання звіту, вибірковий захист	Захист кожної роботи з ШІ-екзаменатором, рішення за викладачем

Таблиця 1: Порівняння підходів. Це наша якісна оцінка, а не виміряні дані.

4. Рішення

Ключова ідея — цікаві студентам завдання і ШІ, який допомагає не тільки студенту, а й викладачу. Виконувати завдання можна будь-якими інструментами, зокрема ШІ, а розуміння перевіряємо в контрольованих умовах: захист обмежений у часі, запитання стосуються саме цієї роботи й заздалегідь невідомі студенту, а вся розмова записується. На online-зустрічі студент працює з увімкненими камерою та демонстрацією екрана. ШІ-екзаменатор аналізує запис і позначає для викладача місця, де відповіді розходяться з роботою.

Чому.Lab складається з чотирьох частин:

1. **Карта компетенцій.** Спільні для всіх результати навчання курсу. Траєкторії студентів різні, а компетенції однакові, тому результати можна порівняти.
2. **Банк реальних задач** різної складності й типу: від коротких вправ до задач від партнерів із реальними даними, командних і польових.
3. **Інтерактивний артефакт замість звіту.** Результат роботи — notebook із моделлю, де можна змінити параметр і одразу побачити наслідок.
4. **ШІ-екзаменатор.** Після здачі студент 10–15 хвилин в аудиторії або під час online-зустрічі відповідає на запитання саме про свою роботу: пояснює рішення, змінює параметр і передбачає результат, шукає помилку у власній моделі. ШІ-екзаменатор пропонує оцінку, а викладач її підтверджує.

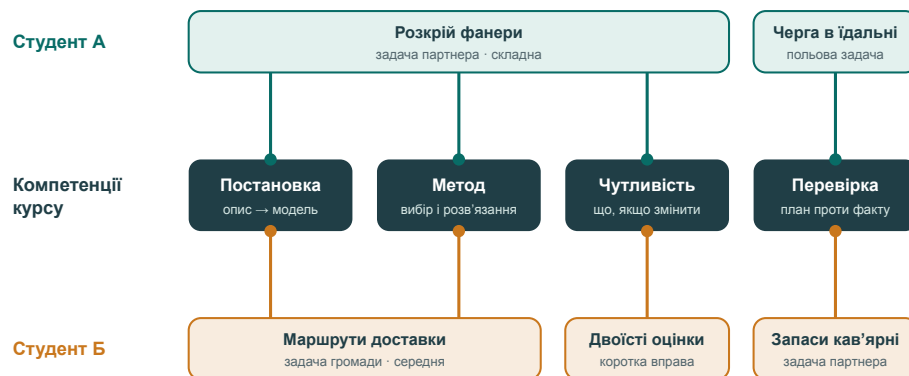


Рисунок 1: Різні задачі, ті самі компетенції. Назви задач умовні: банк задач для пілоту ще складаємо.

5. Як це працює

1. **Студент обирає задачу** з банку: цікаву йому і таку, що закриває потрібні компетенції.
2. **Виконує роботу** вдома будь-якими інструментами, зокрема ШІ. Результат — інтерактивний notebook із моделлю.
3. **Здає роботу.** ШІ-екзаменатор перевіряє роботу й пропонує запитання для захисту, викладач їх затверджує.
4. **Захищає роботу в аудиторії (online-зустрічі).** ШІ-екзаменатор проводить короткий діалог про цю конкретну роботу.
5. **Викладач** переглядає запропоновану оцінку, повний запис діалогу й позначені розбіжності. Остаточне рішення приймає людина.



Рисунок 2: ШІ-екзаменатор проводить діалог із кожним студентом. Викладач пізніше переглядає запис, запропоновану оцінку й позначки та ухвалює рішення.

Приклад: курс «Дослідження операцій». Студент обирає задачу розкрою: як нарізати деталі з листів фанери з найменшими відходами. Він збирає розміри, будує модель цілочисельного програмування, розв'язує її в notebook, вирізає деталі з картону й порівнює план із фактом. На захисті екзаменатор просить змінити розмір листа, спершу передбачити, як зміняться відходи, а потім перевірити запуском.

6. Конкурентна перевага

Окремі складові не нові: проєктне навчання, адаптивні тренажери, усні захисти. Перевага — у їхньому поєднанні:

- **ШІ — інструмент, а не загроза.** Ми не воюємо з ШІ, а перевіряємо те, чого він не замінить: розуміння власної роботи.
- **Свобода вибору при порівнюваному результаті.** Кожен іде своїм шляхом, але всі шляхи ведуть через спільну карту компетенцій.
- **Захист кожної роботи.** Викладач потоку зазвичай не може вислухати кожного студента. ШІ бере на себе діалог, викладач — рішення.
- **Доказ процесу, а не лише результату.** Історія версій, журнал ШІ і запис захисту разом показують, як студент думав.

Будуємо на відкритих інструментах. Власної розробки потребує лише ШІ-екзаменатор.

Чи збігаються оцінки екзаменатора з оцінками викладача, покаже пілот.

7. Стан і наступні кроки

Етап — концепція. Готуємо пілот на курсі «Дослідження операцій» для третього курсу. Результатів ще немає.

1. Скласти карту компетенцій курсу і банк із 10 задач різної складності.
2. Зібрати прототип: робоче середовище студента, ШІ-екзаменатор і панель викладача.
3. Провести пілот на одному потоці протягом семестру.
4. Перевірити три речі: чи збігаються оцінки ШІ й викладача; чи не уникають студенти складних задач; чи не збільшується навантаження на викладача.

8. Кого шукаємо

- **Викладачів** computer science дисциплін, готових спробувати підхід на своєму курсі.
- **Партнерів із реальними задачами:** компанії, лабораторії, громади, які можуть описати свою проблему й поділитися даними.
- **Кафедри й університети,** зацікавлені в пілоті та оцінці його результатів.

9. Запитання та відповіді

Студенти ж усе одно використовуватимуть ШІ?

Так, і це дозволено. Ми перевіряємо не те, хто написав код, а чи розуміє студент свою роботу і чи може її змінити.

Чи ставить ШІ оцінку сам?

Ні. ШІ-екзаменатор пропонує оцінку й показує запис діалогу. Остаточне рішення завжди за викладачем.

Чи не втрачаються фундаментальні знання?

Ні. Базові методи студенти тренують на коротких аудиторних завданнях без ШІ. Без цих знань неможливо оцінити, чи правильну відповідь дав інструмент.

Чи може ШІ визначити, що роботу зробив ШІ?

Ні, такі детектори ненадійні, і ми на них не покладаємося. Розрив між якістю роботи та поясненнями студента — сигнал для викладача, а не вивок.

Навіщо інтерактивний артефакт замість звіту?

Звіт можна лише прочитати. Notebook можна запустити й змінити. Так перевіряється сама модель, а не текст про неї. Notebook водночас слугує звітом.

Що з даними студентів?

Роботи й записи захистів бачать лише студент і викладач. Правила зберігання погодимо з університетом до пілоту.