



Силабус освітнього компонента Програма навчальної дисципліни



Технології програмування

Шифр та назва спеціальності

F5 – Кібербезпека та захист інформації

Інститут

ННІ комп'ютерних наук та інформаційних технологій (320)

Спеціалізація

-

Кафедра

Кібербезпеки (328)

Освітня програма

Кібербезпека

Тип дисципліни

Обов'язкова

Рівень освіти

Перший (бакалаврський)

Форма навчання

Денна, заочна

Семестр

3

Мова викладання

Українська, англійська

Викладачі, розробники

**ТКАЧОВ Андрій Михайлович**

andrii.tkachov@khpi.edu.ua

Кандидат технічних наук, старший науковий співробітник кафедри кібербезпеки НТУ «ХПІ».

Кількість наукових публікацій: більше 60 публікацій, 25 статей у закордонних виданнях та фахових виданнях України, 6 патентів на корисну модель, гарант освітньо-професійної програми "Національна безпека у сфері кіберзахисту" першого (бакалаврського) рівня вищої освіти.

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Навчальна дисципліна "Технології програмування" є обов'язковою навчальною дисципліною. Дисципліна спрямована на формування у студентів теоретико-практичного базису щодо сучасних підходів у сфері програмування та їх впровадження у роботу веб-орієнтованих систем. Вивчення дисципліни забезпечує професійний розвиток та спрямована на дослідження процесів виявлення знань, оволодіння сучасними технологіями та засобами програмування.

Мета та цілі дисципліни

Вивчення технологій програмування, придбання студентами знань і навиків в області розробки алгоритмів, створення, трансляції та налагодження прикладних програм, застосування бібліотек та модулів для створення програмних забезпечень для вирішення задач аналізу та захисту інформаційних систем.

Формат занять

Лекції, лабораторні роботи, самостійна робота, консультації. Підсумковий контроль – іспит.

Компетентності

ЗК1. Здатність застосовувати знання у практичних ситуаціях.

ЗК2. Знання та розуміння предметної області і розуміння професійної діяльності.

ЗК3. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність вчитися і оволодівати сучасними знаннями.

СК1. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі кібербезпеки та захисту інформації.

СК2. Здатність використовувати інформаційні технології, сучасні методи і моделі кібербезпеки та систем захисту інформації.

Результати навчання

РН1. Вільно спілкуватися державною мовою усно та письмово при виконанні професійних обов'язків.

РН4. Організовувати власну професійну діяльність, обирати і використовувати оптимальні методи та способи розв'язання складних спеціалізованих задач і практичних проблем у професійній діяльності, оцінювати їхню ефективність.

РН5. Аналізувати, аргументувати, приймати рішення при розв'язанні складних спеціалізованих задач і практичних завдань у професійній діяльності, які характеризуються комплексністю та неповною визначеністю умов, відповідати за прийняті рішення.

РН6. Адаптуватися до нових умов і технологій професійної діяльності, прогнозувати кінцевий результат.

РН7. Застосовувати й адаптувати теорії інформації та кодування, математичної статистики, чисел, криптографії та стеганографії, оброблення та передачі сигналів тощо, принципи, методи і поняття кібербезпеки та захисту інформації у навчанні та професійній діяльності.

РН8. Застосовувати знання й розуміння математики та фізики в професійній діяльності, формалізувати задачі предметної галузі кібербезпеки та захисту інформації, формулювати їх математичну постановку та обирати раціональний метод вирішення.

РН9. Вміти застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі кібербезпеки та захисту інформації.

РН10. Вміти використовувати сучасні інформаційні технології, методи і моделі кібербезпеки та систем захисту інформації для здійснення професійної діяльності.

Обсяг дисципліни

Загальний обсяг дисципліни 150 год. (5 кредитів ECTS): лекції – 32 год., лабораторні роботи – 32 год., самостійна робота – 86 год.

Передумови вивчення дисципліни (пререквізити)

Основи програмування.

Особливості дисципліни, методи та технології навчання

В ході викладання дисципліни викладачем застосовуються пояснювально-ілюстративний (інформаційно-рецептивний) та репродуктивний методи навчання. В якості методів викладання, які направлені на активізацію та стимулювання навчально-пізнавальної діяльності здобувачів, застосовуються презентації, бесіди, індивідуальні групові проєкти, майстер-класи.

Програма навчальної дисципліни

Навчальні заняття

Лекції

Теми лекцій	Кількість годин
Тема 1. Вступ до курсу. Основні поняття та терміни.	4
Тема 2. Сучасні технології програмування. Огляд та порівняння сучасних парадигм та технологій програмування.	4
Тема 3. Сучасні середовища розробки. Інтегровані середовища розробки програмного забезпечення.	4
Тема 4. Робота з структурами даних. Структури даних. Представлення та використання структур даних.	4
Тема 5. Робота з файлами. Доступ до файлів. Операції з файлами.	4
Тема 6. Основні модулі. Підключення модулів. Використання та компонування модулів.	4
Тема 7. Об'єктно-орієнтоване програмування. Парадигма об'єктно-орієнтованого програмування. Принципи, особливості.	4
Тема 8. Технології контролю версій. Використання технологій контролю версій у сучасних інтегрованих середовищах програмування.	4
Загальна кількість годин	32

Лабораторні заняття

Теми лабораторних занять	Кількість годин	Вагові коефіцієнти α
Тема 1. Вступ до курсу. Предмет і завдання дисципліни. Роль технологій програмування в сучасній IT-індустрії. Поняття життєвого циклу програмного забезпечення. Сучасні підходи до програмування: процедурне, об'єктно-орієнтоване, функціональне.	4	0,1
Тема 2. Сучасні технології програмування. Мови програмування високого рівня (Python, Java, C++, C#). Технології веб-програмування (HTML, CSS, JavaScript, PHP). Хмарні технології та мікросервісна архітектура. Програмування мобільних додатків. Розвиток штучного інтелекту та його використання в розробці.	4	0,1
Тема 3. Сучасні середовища розробки. Інтегровані середовища розробки (IDE): Visual Studio, PyCharm, Eclipse. Текстові редактори для програмістів (VS Code, Sublime Text). Налаштування та тестування програм у середовищах розробки. Розширення та плагіни IDE.	4	0,1

Тема 4. Робота з структурами даних. Лінійні структури: масиви, списки, стеки, черги. Нелінійні структури: дерева, графи. Використання структур даних у стандартних бібліотеках мов програмування. Приклади застосування структур даних у реальних проєктах.	4	0,1
Тема 5. Робота з файлами. Текстові та бінарні файли. Читання, запис та редагування файлів. Серіалізація даних. Використання форматів JSON, XML, CSV. Робота з файловою системою (створення, переміщення, видалення файлів).	4	0,1
Тема 6. Основні модулі. Поняття модульності програмування. Стандартні бібліотеки мов програмування. Розробка та використання власних модулів. Імпорт та організація коду за допомогою пакетів. Приклади застосування модульності у великих проєктах.	4	0,1
Тема 7. Об'єктно-орієнтоване програмування (ООП). Основні принципи ООП: інкапсуляція, наслідування, поліморфізм. Класи та об'єкти. Конструктори та деструктори. Абстрактні класи та інтерфейси. Практичне застосування ООП у сучасних мовах програмування.	4	0,1
Тема 8. Технології контролю версій. Поняття контролю версій. Системи контролю версій: Git, SVN. Основні операції у Git: commit, branch, merge, pull, push. Робота з GitHub, GitLab, Bitbucket. Використання систем контролю версій у командній розробці.	4	0,1

Загальна кількість годин

32

$$\sum_{i=1}^n a_i = 0,8$$

Контрольні роботи

За наявності

Теми контрольних робіт

Вагові
коефіцієнти b

Тема 1. Сучасні технології програмування та середовища розробки. Порівняльна характеристика мов програмування (Python, Java, C++, C#). Сучасні підходи до розробки програмного забезпечення. Роль інтегрованих середовищ розробки (IDE) у процесі створення програм. Приклади використання сучасних технологій у реальних проєктах.	0,5
Тема 2. Об'єктно-орієнтоване програмування та проектування. Основні принципи ООП: інкапсуляція, спадкування, поліморфізм. Класи та об'єкти. Інтерфейси та абстрактні класи. Патерни проектування як основа технології програмування. Приклади застосування ООП у реальних проєктах.	1,5
Загалом	$\sum_{i=1}^m b_i = 2$

Самостійна робота

До самостійної роботи відноситься самостійне опрацювання теоретичного матеріалу.

Опрацювання теоретичного матеріалу

Теми самостійного вивчення

Кількість годин

Технології програмування



Національний технічний університет
«Харківський політехнічний інститут»

Тема 1. Історія розвитку мов програмування.	13
Етапи розвитку: від машинних кодів до мов високого рівня. Процедурне, структурне та об'єктно-орієнтоване програмування. Мови нового покоління (Python, Go, Rust, Kotlin). Вплив історії мов на сучасні парадигми програмування.	
Тема 2. Фреймворки та бібліотеки у сучасному програмуванні.	13
Поняття фреймворку та бібліотеки: відмінності та спільні риси. Приклади серверних фреймворків: Django, Spring, .NET. Клієнтські бібліотеки та фреймворки: React, Angular, Vue. Використання бібліотек для роботи з даними (NumPy, Pandas).	
Тема 3. Паралельне та багатопотокове програмування.	12
Основи багатопотоковості та конкурентності. Потоки, процеси, асинхронність. Використання багатопотоковості у Java, C++, Python. Проблеми синхронізації та управління ресурсами.	
Тема 4. Програмування баз даних та робота з ORM-технологіями.	12
Основи роботи з SQL та NoSQL базами даних. Використання SQL у програмуванні додатків. ORM-технології: Hibernate, SQLAlchemy, Entity Framework. Порівняння традиційних запитів і ORM-рішень.	
Тема 5. Автоматизація тестування та CI/CD у процесі розробки.	12
Види тестування: модульне, інтеграційне, системне. Фреймворки для тестування (JUnit, PyTest, Selenium). Поняття CI/CD та основні інструменти (Jenkins, GitHub Actions, GitLab CI). Автоматизація процесу розгортання програм.	
Тема 6. Хмарні технології у програмуванні (AWS, Azure, Google Cloud).	12
Поняття хмарних обчислень (IaaS, PaaS, SaaS). Використання AWS Lambda, Azure Functions, Google Cloud Run. Контейнеризація та Docker у хмарних середовищах. Інтеграція хмарних сервісів у програмні системи.	
Тема 7. Технології програмування мобільних додатків.	12
Особливості розробки під Android (Java, Kotlin). Особливості розробки під iOS (Swift, Objective-C). Кросплатформні рішення (Flutter, React Native). Тестування та публікація мобільних додатків у маркетах.	
Загальна кількість годин	86

Неформальна освіта

Здобувач має можливість перезарахувати окремі теми або курс шляхом: проходження професійних курсів чи тренінгів, онлайн-освіти, професійних стажувань, у сфері, що відповідає навчальним цілям дисципліни.

Для зарахування необхідно надати: сертифікат (електронний або друкований) про проходження курсу/стажування, опис програми тренінгу із зазначенням змісту тем, обсягу та тривалості.

Рекомендовані курси, тренінги, стажування

1. Онлайн-курс CISCO «Python 1»

<https://www.netacad.com/catalogs/learn?category=course>

Література, навчальні матеріали та інформаційні ресурси

Основна література

1. Python 3.10.2 documentation [Електронний ресурс]. – Режим доступу : <https://docs.python.org/3/>.

2. Java Platform Standard Edition 8 Documentation [Електронний ресурс]. – Режим доступу : <https://docs.oracle.com/javase/8/docs/>.
3. Microsoft C++, C, and Assembler documentation [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/cpp/?view=msvc-170>.
4. WEB-технології [Електронний ресурс]: Навчально-довідковий посібник / С.П. Євсєєв, А.М. Ткачов, В.О. Алексієв, Ю.М. Рябуха – Харків : ХНЕУ ім. С. Кузнеця, – Львів: Видавництво «Новий Світ – 2000», 2021. – 390 с.
<https://drive.google.com/drive/u/1/folders/1wOTN8N-GBGO06AnvjQHU1SdBl3xCaUju>
5. Євсєєв С.П. КІБЕРБЕЗПЕКА: КРИПТОГРАФІЯ З PYTHON: навч. посібн. / С.П. Євсєєв, О.В. Шматко, О.Г. Король – Харків: Видавництво «Новий Світ – 2000», 2021. – 120 с.
<https://drive.google.com/drive/u/1/folders/1wOTN8N-GBGO06AnvjQHU1SdBl3xCaUju>

Додаткова література

6. Python Tutorial [Електронний ресурс]. – Режим доступу : <https://www.tutorialspoint.com/python/index.htm>.
7. Java Tutorial [Електронний ресурс]. – Режим доступу : <https://www.tutorialspoint.com/java/index.htm>.
8. C++ Tutorial [Електронний ресурс]. – Режим доступу : <https://www.tutorialspoint.com/cplusplus/index.htm>.

Система оцінювання

Підсумкова оцінка з освітнього компонента визначається відповідальним лектором за темами, видами занять, тощо у відповідності до силабусу і є інтегральною оцінкою результатів усіх вид навчальної діяльності здобувача вищої освіти. Підсумкова оцінка повинна відображати всі оцінки за складовими навчального процесу з урахуванням їх вагових показників k :

Поточний контроль (практичні, семінарські, лабораторні заняття), k_1	Контрольні роботи (за наявності), k_2	Індивідуальне завдання (за наявності), k_3	Підсумковий контроль (для ОК з іспитом), k_4
0,4	0,2		0,4

Сума коефіцієнтів повинна складати одиницю: $k_1 + k_2 + k_3 + k_4 = 1$. Підбір вагових коефіцієнтів підсумкової оцінки здійснює розробник курсу.

Розрахунок підсумкової оцінки проводиться за формулою:

$$O = \Pi \cdot k_1 + K \cdot k_2 + I \cdot k_3 + \Pi_k \cdot k_4$$

де: Π – середньозважена середня оцінка за поточний контроль

I – оцінка за виконання індивідуального завдання

K – середньозважена оцінка за контрольні роботи

Π_k – оцінка за підсумковий контроль

$$\Pi = \frac{\Pi_1 \cdot a_1 + \Pi_2 \cdot a_2 + \dots + \Pi_n \cdot a_n}{\sum_{i=1}^n a_i}$$

де: a_i - ваговий коефіцієнт за кожне практичне (семінарське) або лабораторне заняття.

$$K = \frac{K_1 \cdot b_1 + K_2 \cdot b_2 + \dots + K_m \cdot b_m}{\sum_{i=1}^m b_i}$$

де: b_i - ваговий коефіцієнт за кожну контрольну роботу.

Шкала оцінювання

Поточні оцінки за кожну складову (П, К, І,...) виставляються за 100-бальною шкалою згідно з положенням «Про критерії та систему оцінювання знань та вмінь і про рейтинг здобувачів вищої освіти» НТУ «ХПІ».

Підсумкова оцінка виставляється відповідно до розрахованої О з округленням до найближчого цілого числа в більшу сторону.

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Здобувач вищої освіти повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту.

Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

30.08.2025



Завідувач кафедри
Сергій ЄВСЄЄВ

30.08.2025



Гарант ОП
Сергій ЄВСЄЄВ