

 <p>ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ</p>	<p>Кафедра комп'ютерної інженерії та інформаційних технологій</p> <p>СИЛАБУС</p>
<p>Базова інформація про дисципліну</p>	
<p>Назва дисципліни</p>	<p>СЕ129 / Програмування / Programming</p>
<p>Рівень вищої освіти / фахової передвищої освіти</p>	<p>Перший (бакалаврський)</p>
<p>Галузь знань</p>	<p>12 Інформаційні технології</p>
<p>Спеціальність</p>	<p>123 «Комп'ютерна інженерія»</p>
<p>Освітня програма</p>	<p>Комп'ютерна інженерія</p>
<p>Семестр</p>	<p>1 семестр</p>
<p>Курс</p>	<p>1 курс</p>
<p>Анотація курсу</p>	<p>Курс «Програмування» забезпечує фундаментальні знання та практичні навички, необхідні для розробки програмного забезпечення. Студенти знайомляться з основними етапами життєвого циклу розробки програм — від проектування до тестування та підтримки, а також із типовими архітектурними підходами у програмуванні. Курс надає загальну інформацію про сучасні мови програмування, їхні екосистеми та принципи вибору відповідної мови для конкретних завдань.</p> <p>Особлива увага приділяється використанню систем контролю версій, зокрема Git, для управління змінами у програмному коді. Студенти вивчають основи роботи з гілками та основні операції Git, а також отримують навички роботи з платформою GitHub для спільної розробки проєктів.</p> <p>Курс також охоплює інструменти, які використовуються в сучасному програмуванні, включно з інтегрованими середовищами розробки (IDE) та редактором Visual Studio Code. Окремий акцент робиться на можливостях хмарної розробки програм та використанні вбудованих в редактори коду інструментів штучного інтелекту для прискорення написання програми.</p> <p>Програмування на мові Java займає центральне місце в курсі. Студентам даються основи синтаксису мови програмування Java, правила написання чистого та зрозумілого коду, роботи з масивами, операторами</p>

	<p>контролю керування програмою, тощо. Крім цього, курс знайомить з платформою Java: роботою з файловою системою, основними колекціями, а також з особливостями функціонального програмування в Java, зокрема лямбда-виразами та операціями з потоками даних.</p> <p>Окрема тема присвячена принципам написання чистого коду та покращення якості програмного забезпечення. Розглядаються базові принципи чистого коду, такі як DRY, KISS і YAGNI.</p>
Сторінка курсу в MOODLE	http://78.137.2.119:2929/course/view.php?id=540
Мова викладання	Українська
Лектор курсу	<p>Ночевнов Дмитро Павлович, доцент, к.т.н.</p> <p>канали комунікації: СДН «Moodle»: повідомлення в чаті E-mail: dmitry.ndp@gmail.com</p>
Місце дисципліни в освітній програмі	
Освітня програма	http://csbc.edu.ua/documents/otdel/oop_kb1.pdf
Перелік загальних компетентностей (ЗК)	<p>Z1. Здатність до абстрактного мислення, аналізу і синтезу. Z2. Здатність вчитися і оволодівати сучасними знаннями. Z3. Здатність застосовувати знання у практичних ситуаціях.</p> <p>Z4. Здатність спілкуватися державною мовою як усно, так і письмово.</p> <p>Z5. Здатність спілкуватися іноземною мовою.</p> <p>Z6. Навички міжособистісної взаємодії.</p> <p>Z7. Вміння виявляти, ставити та вирішувати проблеми.</p> <p>Z8. Здатність працювати в команді.</p>
Перелік спеціальних компетентностей (СК)	<p>P2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.</p> <p>P3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.</p> <p>P11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.</p> <p>P13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.</p>
Перелік програмних результатів навчання	<p>N3. Знати новітні технології в галузі комп'ютерної інженерії.</p> <p>N10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.</p> <p>N11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.</p>

	N12. Вміти ефективно працювати як індивідуально, так і у складі команди.
Опис дисципліни	
Структура навантаження на студента	Загальна кількість годин – 180 Кількість кредитів – 6 Кількість лекційних годин – 30 Кількість практичних занять – 30 Кількість годин для самостійної роботи студентів – 120 Форма підсумкового контролю – іспит
Методи навчання	Словесні (лекція, самостійна робота з джерелами інформації, науково-популярна розповідь); Наочні (презентаційні повідомлення) Практичні (лабораторні роботи); Інтерактивні методи (дистанційні консультації).
Зміст дисципліни	
Тема 1. Основні складові програмування	Етапи і засоби розробки програмного забезпечення: проектування, розробка, побудова, запуск, відлагодження, тестування, підтримка. Типова архітектура програмного забезпечення. Критерії вибору мови програмування Екосистема мови програмування.
Тема 2. Системи контролю версій програмного коду	Призначення і історія створення систем контролю версій. Складові частини Git. Етапи зміни стану файлу в Git. Використання гілок в Git. Операції Git. Функціональність GitHub.com
Тема 3. Інструментарій програмування. Використання штучного інтелекту	Основні функції IDE - Integrated Development Environment. Редактор програмного коду Visual Studio Code. Редактор програмного коду IntelliJ IDEA. Розробка програм в хмарному середовищі. Робота з вбудованими AI-помічниками.
Тема 4. Основи синтаксису мови програмування Java	Особливості синтаксису мови Java. Домовленості по найменуванню в Java. Коментарі. Вирази і операції. Примітивні типи даних і класи-обгортки. Оператори контролю виконання програми. Масиви. Синтаксис методів. Анотації в Java.
Тема 5. Робота з платформою Java	Введення / виведення даних. Робота з файловою системою. Робота з колекціями.
Тема 6. Функціональне програмування в Java	Лямбда-вирази. Потоки даних. Потоківі операції з масивами і колекціями.
Тема 7. Чистий код	Базові принципи чистого коду DRY, KISS, YAGNI. Чистий код класів, інтерфейсів, методів і функцій, змінних, операторів. Форматування коду. Самодокументуючий код. Статичні аналізатори чистоти коду.
Політика дисципліни	

Політика відвідування	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання зорганізується в онлайн формі за погодженням з керівником курсу.
Політика щодо дедлайнів та перескладання	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
Академічна доброчесність	У випадку недотримання політики академічної доброчесності (плагиат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.

Система оцінювання

Поточний контроль здійснюється протягом семестру під час проведення лабораторно-практичних занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/семестровий контроль, проводиться у формі іспиту, відповідно до графіку навчального процесу.

Накопичування рейтингових балів з навчальної дисципліни

Види навчальної роботи	Мах кількість балів
Виконання лабораторних робіт № 1-4	40
Модульні контрольні роботи (2 к.р.)	20
Індивідуальне завдання	10
Завдання іспиту	30
Разом	100

Шкала оцінювання

ECTS	Бали	Зміст
A	90-100	Бездоганна підготовка в широкому контексті
B	80-89	Повні знання, міцні вміння
C	70-79	Хороші знання та вміння
D	65-69	Задовільні знання, стереотипні вміння
E	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
FX	35-59	Слабкі знання, відсутність умінь
F	1-34	Необхідний повторний курс

Список рекомендованих джерел

Основна

1. Дэвид Флэнаган, Бенджамин Эванс. Java. Справочник разработчика. - Диалектика, 2019. - 592 с.
2. Кетті Сьерра, Берт Бейтс. Head First Java. Легкий для сприйняття довідник. - Видавництво : Фабула, 2022, - 720 с.
3. Раджу Ганди. Head First. Git. - Видавництво: Print2print, 2023, - 464 с.
4. Роберт Мартин. Чистий код. - Видавництво: #PROSystem, 2019, - 448 с.
5. Рауль-Габриэль Урма, Марио Фуско, Алан Майкрофт. Сучасна мова Java. Лямбда-вирази, потоки та функціональне програмування. - Видавництво: Print2print, 2020, - 592 с.

Додаткова

1. [GitHub Education](#).
2. [Scott Chacon, Ben Straub. Pro Git. - Apress, 2014, - 426 p.](#)
3. [Tutorial: Get started with Visual Studio Code](#)
4. [GitHub Copilot documentation](#)
5. [The Java™ Tutorials](#)