

СИЛАБУС

Кафедра кібербезпеки та математичного моделювання

Назва курсу	Безпечні технології програмування
Мова викладання	українська
Викладач	Базилевич В.М., к.е.н., доцент, зав. кафедри інформаційних та комп'ютерних систем
Профайл викладача	Web: https://cs.stu.cn.ua/ ResearcherID: G-5764-2014 ORCID: 0000-0001-8935-446X
Контакти викладача	bazvlamar@stu.cn.ua

1. Анотація курсу –

Змістовий модуль 1. Класи та об'єкти

Тема 1. Вступ. Складність програмного забезпечення

Предмет та завдання курсу “Безпечні технології програмування”. Структура навчального курсу. Навчально-методична література з дисципліни. Терміни і означення. Основна проблематика ООП. Складність програмного забезпечення та засоби боротьби з нею. Абстракція, декомпозиція, ієрархія. Історія розвитку ООП та особливості окремих мов програмування.

Тема 2. Поняття класу

Поняття класу та об'єкту. Стан об'єкту та правила поведінки. Структура об'єкта та його характеристики. Структура класу. Особливості та приклади визначення класу об'єктів в мовах *Smalltalk*, *Object Pascal*, *C++*, *Java*. Класові типи *C++*. Шаблони класів. Класові посилки. Модифікатори класів та визначення інтерфейсів в *Java*. Пакети класів та порядок їх створення. Правила виконання програм *Java*.

Тема 3. Поля даних та їх ініціалізація

Різновиди стану об'єктів, їх статика та динаміка. Логічна структура об'єктів. Види змінних об'єктів. Змінні класу та змінні екземпляру. Визначення змінних в *Smalltalk*, *C++*, *Object Pascal*, *Java*. Поля даних в *C++*. Особливості визначення статичних змінних. Специфікатори доступу до даних та правила їх використання. Спеціальні змінні *self*, *super*, *this*, *sender*. Динамічні та статичні об'єкти. Посилки та особливості їх визначення.

Тема 4. Конструктори та деструктори

Створення об'єктів та їх знищення в *Smalltalk*, *C++*, *Object Pascal*, *Java*. Індивідуальність об'єктів. Копіювання та порівнювання об'єктів. Інспектування об'єктів. Особливості використання спеціальних змінних *self*, *this* та *super*. Різновиди конструкторів в *C++* та особливості їх визначення. Правила ініціалізації об'єктів. Порядок виклику конструкторів та деструкторів. Перевантаження конструкторів. Особливості виконання процедур “збору сміття” в різних мовах програмування.

Тема 5. Методи

Різновиди операцій над об'єктами. Загальна схема методу. Методи класу та методи екземпляру. Особливості визначення методів в *Smalltalk*, *C++*, *Object Pascal*, *Java*. Перевантаження та заміщення методів. Правила визначення вміщених методів (*inline*). Посилки на клас в *Delphi* та приклади їх використання. “Дружні” методи в *C++* та їх реалізація в *Object Pascal* через вказівки на методи. Функції-операції в *C++* та їх реалізація. Правила перевизначення операцій. Абстрактні методи.

Змістовий модуль 2. Основні концепції ООП

Тема 6. Інкапсуляція абстракцій

Поняття інкапсуляції. Організація безпеки доступу до властивостей об'єктів.

Властивості в *Delphi* та особливості їх визначення. Внутрішнє збереження властивостей та обмеження в їх використанні. Інтерфейс та реалізація.

Тема 7. Спадкування

Форми та способи спадкування. Ієрархія класів. Спадкування в *Smalltalk*: метакласи та їх ієрархія, механізм виклику методів. Спадкування в C++: специфікатори спадкування та правила доступу в класі, покажчик на базові класи, правило сумісності типів, віртуальне спадкування та його схема, особливості використання конструкторів та деструкторів при спадкуванні. Спадкування в *Delphi*: особливості реалізації спадкування, виклики захищених методів, ієрархія класів, схема розробки нових класів, порядок створення нових компонент. Спадкування в *Java*: особливості спадкування, спадкування інтерфейсів, ієрархія класів та пакетів.

Тема 8. Поліморфізм

Поняття поліморфізму. Віртуальні методи. Особливості використання віртуальних функцій в C++: механізм пізнього зв'язування, схема використання VMT. Поліморфізм в *Delphi*: віртуальні та динамічні методи, RTTI та її формат, приклади використання RTTI, правила заміщення віртуальних та динамічних методів, схема поліморфізму, порядок приведення типів, оператори *is* та *as*, засоби обробки виключних ситуацій. Особливості реалізації поліморфізму в *Java* та *SmallTalk*: віртуальна машина та інтерпретатор байт-коду, схема пізнього зв'язування.

Змістовий модуль 3. Практичне використання ООП парадигми у Java

Тема 9. Примітивні типи даних та масиви Java

Вирішення проблеми примітивних типів. . Примітивні типи даних, класи обгортки. Аутобоксінг, анбоксінг. Масиви, створення та ініціалізація масивів. Багатовимірні масиви.

Тема 10. Внутрішні класи Java

Визначення внутрішнього класу. Статичні та нестатичні внутрішні класи. Локальні класи та безіменні класи. Реалізація проблеми передачі специфічних операцій користувача до узагальнених методів.

Тема 11. Стандартна бібліотека класів Java

Класи візуальних компонент. Огляд менеджерів розташування компонентів. Утилітні класи *System*, *Math*, *Arrays*, *Collections*. Класи, що забезпечують графічні можливості мов програмування.

Тема 12. Контейнерна бібліотека Java

Огляд контейнерної бібліотеки. Огляд класів колекцій, списків, множин та асоційованих масивів. Ітератори та засоби їх реалізації

Тема 13. Робота з файлами в Java

Огляд бібліотеки вводу виводу Поточкові класи: форматне введення-виведення, маніпулятори, обмін інформацією з файлами.

Тема 14. Багатопоточне програмування в Java

Поняття процесу і потоку. Класи, що забезпечують можливості реалізації паралелізму. Способи створення потоків. Засоби керування взаємодією потоків. Проблема сумісного використання даних. Синхронізація. Рівні безпечності потоків. Безпечне багатопоточне програмування

Тема 15. Реалізація механізму подій в Java

Проблема взаємозв'язків між джерелом події та її слухачем. Механізм реалізації події на боці джерела та її обробки на боці слухача. Стандартні класи та інтерфейси для реалізації подій.

Тема 16. Підсумки

Огляд тем курсу. Перспективи використання безпечних технологій програмування для вирішення наукових і практичних задач.

2. Мета та цілі курсу -

Метою викладання навчальної дисципліни “*Безпечні технології програмування*” є закріплення та розвиток загальних та фахових компетентностей магістра в галузі знань 12 – *Інформаційні технології*.

Зокрема, це такі загальні компетентності, як:

КЗ 1. Здатність застосовувати знання у практичних ситуаціях;

КЗ 2. Знання та розуміння предметної області та розуміння професії;

КЗ 4. Вміння виявляти, ставити та вирішувати проблеми за професійним спрямуванням;

КЗ 5. Здатність до пошуку, оброблення та аналізу інформації.

Також такі фахові компетенції:

КФ 2. Здатність розробляти, впроваджувати і супроводжувати програмні та програмно-апаратні комплекси засобів інформаційної безпеки та/або кібербезпеки в інформаційно-комунікаційних системах (автоматизованих систем та їх додатків) та у інфраструктурі організації в цілому.

КФ 3. Здатність до проектування, впровадження, супроводження інформаційних мереж і ресурсів, інфраструктури установи, архітектури використання інформаційних технологій (хмарних), а також бізнес/операційних процесів з метою якісного функціонування інформаційно-комунікаційних систем (комутативних або без комутативних), згідно встановленої політики інформаційної безпеки та/або кібербезпеки і стратегії організації.

КФ 7. Здатність розробляти, впроваджувати, супроводжувати системи аудиту та моніторингу якості бізнес/операційних процесів функціонування організації та системи управління інформаційною безпекою та/або кібербезпекою за вітчизняними і світовими нормами та стандартами.

3. Результати навчання:

Під час вивчення дисципліни ЗВО має досягти або вдосконалити наступні програмні результати навчання (ПРН), передбачені освітньою програмою:

ПРН 2. планувати та організовувати власну професійну діяльність, обирати оптимальні методи та способи розв’язування складних спеціалізованих задач та практичних проблем у професійній діяльності, оцінювати їхню ефективність;

ПРН 3. використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності;

ПРН 4. аналізувати, аргументувати, приймати рішення при розв’язанні складних спеціалізованих задач та практичних проблем у професійній діяльності, які характеризуються комплексністю та неповною визначеністю умов, відповідати за прийняті рішення;

ПРН 5. реалізовувати процеси постійної самоосвіти і професійної сертифікації, критично осмислювати основні теорії, принципи, методи і поняття у навчанні та професійній діяльності;

ПРН 8. проектувати, впроваджувати, супроводжувати системи та комплекси (програмні, програмно-апаратні) захисту додатків (веб - додатків) з метою забезпечення якісного функціонування інформаційно-комунікаційних систем, згідно встановленої політики інформаційної безпеки та/або кібербезпеки і стратегії організації;

ПРН 11. розробляти, впроваджувати, супроводжувати системи аудиту та моніторингу (контролю) якості бізнес/операційних процесів функціонування організації та системи управління інформаційною безпекою та/або кібербезпекою за вітчизняними і світовими нормами та стандартами;

ПРН 12. розробляти, впроваджувати та супроводжувати процеси належного функціонування системи моніторингу інформаційних ресурсів і бізнес процесів в інфраструктурі організації;

ПРН 13. проводити та планувати навчання персоналу компанії, користувачів з інформаційних технологій організації у відповідності до сучасних норм, вимог, внутрішніх правил безпечного застосування інформаційних технологій, а також у відповідність вітчизняним і світовим стандартам галузі інформаційної та/або кібербезпеки;

У результаті вивчення навчальної дисципліни студент повинен:

знати :

1. базові поняття ООП – клас, об'єкт, метод
2. основні концепції ООП – абстракція, інкапсуляція, спадковість, поліморфізм;
3. застосовувати інкапсуляцію для контролю атрибутів, і безпеки класів
4. особливості визначення класів;
5. порядок ініціалізації об'єктів;
6. види спадкування;
7. механізм реалізації поліморфізму;
8. методика використання стандартних бібліотек класів.

вміти :

1. визначати класи об'єктів;
2. регламентувати доступ до даних і методів
3. реалізовувати методи;
4. визначати ієрархії класів;
5. використовувати стандартні бібліотеки.
6. використовувати технологію reflection для контролю атрибутів класу

4. Обсяг курсу. 3 кредитів ECTS, що становить 90 годин роботи студентів, з них 60 годин самостійної роботи та 30 годин аудиторної роботи з викладачем.

Вид заняття	Загальна кількість годин
Лекції	16
Лабораторні заняття	14
Самостійна робота (РГР, наукові дослідження)	60

5. Пререквізити. Передумовою для вивчення дисципліни “Безпечні технології програмування” є успішне засвоєння дисциплін “Інформатика”, “Технології програмування” та здобуті такі результати навчання, як поняття алгоритму, вміння складати програми мовою С, знання персонального комп'ютера.

Набуті під час вивчення дисципліни “Безпечні технології програмування” знання та вміння застосовуються у виконанні випускної магістерської роботи.

Обов'язковою умовою викладання дисципліни є проведення лабораторного практикуму із застосуванням персональних комп'ютерів та спеціалізованого інтегрованого середовища розробки програм, наприклад, Eclipse.

6. Система оцінювання та вимоги

Загальна система оцінювання курсу	ECTS
Вимоги розрахунково-графічної роботи	При перевірці та оцінюванні розрахунково-графічної роботи враховується правильність виконання теоретичних та практичних завдань, самостійність виконання, вчасність здачі роботи та відповідність оформлення результатів діючим вимогам
Лабораторні заняття	Кожна виконана лабораторна робота оцінюється від 0 до 3-х балів. Кількість балів залежить від рівня теоретичних знань та практичних навичок студента за темою, самостійності виконання роботи та вчасності її захисту
Умови допуску до підсумкового контролю	Умовою допуску до екзамену є виконання та отримання хоча б мінімальної кількості балів з усіх обов'язкових видів навчальної роботи передбачених робочою програмою (лабораторних, модульного контролю та розрахунково-графічної роботи). Мінімальна кількість балів необхідна для допуску до екзамену/заліку – 20.

Поточний контроль за результатами лабораторних робіт

		Кількість балів	
Усього		0...	100
1	Підготовленість до лабораторної роботи.	0...	10
2	Самостійність виконання лабораторної роботи.	0...	25
3	Своєчасність виконання лабораторної роботи.	0...	15
4	Повнота і якість оформлення звіту.	0...	6
5	Теоретичне питання	0...	22
6	Практичне завдання	0...	22

Підсумковий контроль

Іспит		Кількість балів	
1	Теоретичне питання із практичним завданням по темі питання	0...	12
2	Теоретичне питання із практичним завданням по темі питання	0...	12
3	Практичне завдання із практичним завданням по темі питання	0...	12
4	Додаткові питання	0...	4
Оцінка за іспит		0...	40

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену	
90 – 100	A	відмінно	
82-89	B	добре	
74-81	C		
64-73	D		
60-63	E	задовільно	
0-59	FX	незадовільно з можливістю повторного складання	

7. Політики курсу.

7.1 Академічна доброчесність – самостійність виконання навчальних завдань та посилання на джерела у випадку використання напрацювань інших авторів. Види

порушень академічної доброчесності – академічний плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво.

Відповідно до Положення про академічну доброчесність студентів та науково-педагогічних працівників Національного університету «Чернігівська політехніка» Затв. Вченою радою НУ «Чернігівська політехніка» 31 серпня 2020 р. протокол № 6 Введено в дію наказом ректора від 31 серпня 2020 р. № 26 за порушення академічної доброчесності здобувачі освіти можуть мати наслідком: повторне проходження оцінювання (контрольна робота, іспит, залік тощо); повторне проходження відповідного освітнього компонента освітньої програми; відрахування із закладу освіти (крім осіб, які здобувають загальну середню освіту); позбавлення академічної стипендії; позбавлення наданих закладом освіти пільг з оплати навчання.

7.2 Політика дедлайнів – своєчасність здачі лабораторної роботи оцінюється в 0,5 бала за кожен лабораторну роботу. Своєчасність здачі РГР оцінюється в 2 бали. Відповідно, максимальна оцінка за невчасно здані роботи зменшується на зазначену кількість балів. Виключенням може бути наявність поважних причин несвоєчасної здачі зазначених робіт (хвороба, участь в зазначений час в інших видах навчальної, наукової чи організаційної роботи, офіційна робота за фахом).

7.3 Політика перезарахування кредитів у випадку мобільності – перезарахування відбувається якщо назви навчальних дисциплін ідентичні або мають незначну стилістичну відмінність, але обсяги та змістова частина навчальних програм не відрізняються; кількість кредитів, відведена на вивчення навчальної дисципліни відрізняється менше, ніж на 25 %; форми підсумкового контролю з дисциплін однакові. При перезарахуванні дисципліни зберігається раніше здобута позитивна оцінка. Перескладання іспиту з дисципліни з метою підвищення оцінки, визначеної в документах виданих здобувачу вищої освіти за попереднім місцем навчання, не дозволяється. Перезарахування кредитів проводиться відповідно до Порядку визначення академічної різниці та перезарахування навчальних дисциплін у Національному університеті «Чернігівська політехніка» Затв. Вченою радою НУ «Чернігівська політехніка» 31 серпня 2020 р. протокол № 6 Введено в дію наказом ректора від 31 серпня 2020 р. № 26.

7.4 Політика щодо відвідування – відвідування занять є обов'язковим. При наявності поважних причин (хвороба, участь в інших видах навчальної, наукової чи організаційної роботи, офіційна робота за фахом) студенти можуть узгодити з викладачем індивідуальний графік навчання та здачі всіх видів навчальної роботи. Студенти можуть перескладати або відпрацьовувати пропущені заняття на консультаціях викладача чи у спеціально відведений викладачем для цього час.

7.5 Політика щодо правил поведінки на заняттях – активна участь у навчальному процесі, виконання необхідного мінімуму навчальної роботи, коректна поведінка щодо інших учасників навчального процесу, взаємоповага, використання мобільних пристроїв тільки для навчання.

7.6 Політика заохочень та стягнень. Результати навчальної, наукової та організаційної діяльності студентів за напрямами курсу їм можуть нараховуватися додаткові бали - до 10 балів, в залежності від вагомості досягнень студента. Види позанавчальної діяльності, за які студенти заохочуються додатковою кількістю балів: участь у міжнародних проектах, наукові дослідження, тези, статті на науково-практичних конференціях, винаходи, патенти, авторські свідоцтва за напрямами курсу.

8 Методичне забезпечення

1. Об'єктно орієнтоване програмування на Java. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Об'єктно орієнтоване програмування» для студентів спеціальності 123 – “Комп'ютерна інженерія”. /Укл.: Бивойно П.Г., Бивойно Т.П. – Чернігів: ЧНТУ, 2018. – 108 с..
2. Реалізація поліморфізму засобами Java. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Об'єктно-орієнтоване програмування» для студентів напряму підготовки 6.050102 – „Комп'ютерна інженерія”, частина друга. /Укл.: П.Г. Бивойно, Т.П. Бивойно – Чернігів: ЧДТУ, 2015. – 59 с.
3. Колекції та потоки в Java. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Об'єктно-орієнтоване програмування» для студентів напряму підготовки 6.050102 – „Комп'ютерна інженерія”, частина третя. /Укл.: П.Г. Бивойно, Т.П. Бивойно – Чернігів: ЧДТУ, 2015. – 43 с.
4. Створення деревовидних структур даних засобами Java. Методичні вказівки до виконання розрахунково-графічної роботи з дисципліни «Об'єктно-орієнтоване програмування» для студентів напрямів підготовки 6.050102 – „Комп'ютерна інженерія”, 6.050103 – „Програмна інженерія”. /Укл.: Бивойно П.Г., Бивойно Т.П. – Чернігів: ЧНТУ, 2015. – 28 с.

Рекомендована література

Базова

1. Бадд Т. Объектно-ориентированное программирование в действии. Санкт-Петербург: Питер, 1997.
2. Буч Г. Объектно-ориентированное проектирование с примерами применения: Пер. с англ. – М.:Кокорд,1992. – 519с
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++: Пер. с англ. – М.:Диалект,1999
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. – СПб.:Питер,2001
5. Фаулер Мартин Архитектура корпоративных программных приложений. СПб.: Издательский дом «Вильямс», 2007. – 544с
6. Simon Kendal. Object oriented programming using Java. Ventus Publishing ApS-, 2009. – 209 с.

Допоміжна

1. Дэвид Флэнэген. Java in a Nutshell: пер. с англ. – К.: Издательская группа ВHV, 1998. – 720 с.
2. Волш А.И. Основы программирования на Java для World Wide Web. – К.: Диалектика, 1996. – 512 с.
3. Эккель Б. Философия Java. Библиотека программиста. – СПб.: Питер, 2003
4. Блох Д. Java. Эффективное программирование. – Лори (SUN published), 2003
5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил.
6. Стелтинг Стивен, Маасен Олав. Применение шаблонов Java. Библиотека профессионала. : Пер. с англ. — М.: Издательский дом "Вильямс", 2002. — 576 с.: ил.

Інформаційні ресурси

1. <https://docs.oracle.com/javase/tutorial/>
2. <https://programming.in.ua/programming/java.html>
3. <https://eln.stu.cn.ua/course/view.php?id=749>