

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДВНЗ «ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»**

Факультет математики та інформатики

Кафедра диференціальних рівнянь і прикладної математики

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**«Системне програмування та спеціалізовані мови програмування»**

Освітня програма «Прикладна математика»

Спеціальність 113 «Прикладна математика»

Галузь знань 11 «Математика і статистика»

Денна форма навчання

Затверджено на засіданні кафедри  
Протокол № 2 від 28 вересня 2020 р.

## ЗМІСТ

1. Загальна інформація
2. Анотація до курсу
3. Мета та цілі курсу
4. Компетентності
5. Результати навчання
6. Організація навчання курсу
7. Система оцінювання курсу
8. Політика курсу
9. Рекомендована література

<b>1. Загальна інформація</b>	
<b>Назва дисципліни</b>	Системне програмування та спеціалізовані мови програмування
<b>Рівень вищої освіти</b>	перший (бакалаврський)
<b>Викладач (-і)</b>	Махней Олександр Володимирович
<b>Контактний телефон викладача</b>	(0342)596027
<b>Е-mail викладача</b>	makhney1@yahoo.com
<b>Формат дисципліни</b>	лекції, лабораторні заняття
<b>Обсяг дисципліни</b>	6 кредитів ЄКТС
<b>Посилання на сайт дистанційного навчання</b>	www.d-learn.pnu.edu.ua
<b>Консультації</b>	четвер 13:30
<b>2. Анотація до курсу</b>	
<p>Навчальна дисципліна «Системне програмування та спеціалізовані мови програмування» є нормативною дисципліною для підготовки бакалавра зі спеціальності «Прикладна математика». Знання, набуті студентами з цієї дисципліни, можуть застосовуватись майбутніми фахівцями для програмування паралельних процесів і потоків та розширеного доступу до файлів. Викладання дисципліни ґрунтується на програмуванні в C/C++ з використанням функцій Windows API.</p>	
<b>3. Мета та цілі курсу</b>	
<p><b>Мета</b> викладання навчальної дисципліни: ознайомити студентів із використанням функцій Windows API для програмування паралельних процесів і потоків та розширеного доступу до файлів в C/C++.</p> <p><b>Завдання</b> вивчення навчальної дисципліни: навчити студентів користуватись найважливішими функціями Windows API для роботи з процесами, потоками, файлами і пам'яттю, сформувати уявлення про програмування на асемблері.</p>	
<b>4. Компетентності</b>	
<p>Знання основних мов програмування та інструментальних програмних засобів, що призначені для реалізації алгоритмів, здатність проектувати бази даних, інформаційні системи та ресурси.</p> <p>Здатність оволодіти сучасними технологіями програмування та тестування програмного забезпечення.</p> <p>Здатність застосовувати методи програмування при розробці інформаційних систем, визначати структури даних при проектуванні алгоритмів у процесі вирішення задач загальної природи.</p> <p>Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.</p>	
<b>5. Результати навчання</b>	
<p>Самостійно працювати над дослідницькою темою, обґрунтовувати і створювати програмну реалізацію розроблених методів.</p> <p>Уміти застосовувати методи програмування при розробці інформаційних систем та визначати структури даних при проектуванні алгоритмів у процесі вирішення задач.</p> <p>Комплексно володіти основними методами розробки програмного забезпечення, основ управління проектами розробки програмного забезпечення, основ забезпечення та контролю якості програмного забезпечення.</p> <p>Уміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.</p>	
<b>6. Організація навчання курсу</b>	
Обсяг курсу	
Вид заняття	Загальна кількість годин
лекції	30
семінарські заняття / практичні / лабораторні	30
самостійна робота	120

Ознаки курсу					
Семестр	Спеціальність	Курс (рік навчання)		Нормативний / вибірковий	
7	113 Прикладна математика	4		нормативний	
Тематика курсу					
Тема, план	Форма заняття	Література	Завдання, год.	Вага оцінки, %	Термін виконання
Тема 1. <b>Системне програмування і структура операційної системи Windows</b> Поняття про системне програмування. Основні можливості операційних систем. Структура операційної системи Windows. Підсистема Windows API.	лекція	3, с. 7–40	Опрацювати літературу, прочитати конспект (4 год.)	0,3	вересень
Тема 2. <b>Використання файлової системи і функцій введення/виведення Windows</b> Операції відкриття, читання, запису і закриття файлів. Використання символів Unicode. Керування файлами і каталогами.	лекція	6, с. 51–83	Опрацювати літературу, прочитати конспект (4 год.)	0,3	вересень
Тема 3. <b>Спеціальні засоби для роботи з файлами і каталогами</b> 64-бітна файлова система. Вказівники файлів. Визначення розміру файлу. Атрибути файлів і керування каталогами. Блокування файлів.	лекція	6, с. 87–112	Опрацювати літературу, прочитати конспект (4 год.)	0,3	вересень
Тема 4. <b>Читання і запис файлів засобами C++</b> Потоки введення з текстового файлу і виведення в файл. Зміна позиції читання (запису) у файлі. Обробка тексту.	лабораторне заняття	1	Виконати лабораторну роботу (4 год.)	2,7	вересень
Тема 5. <b>Обробка особливих ситуацій</b> Особливі ситуації та їхня обробка. Генерація програмою особливих ситуацій. Блок завершення. Особливі ситуації при операціях над числами з плаваючою крапкою.	лекція	6, с. 126–153	Опрацювати літературу, прочитати конспект (4 год.)	0,3	вересень
Тема 6. <b>Читання і запис файлів засобами Windows API (частина 1)</b>	лабораторне заняття	1, 2, 6	Виконати лабораторну роботу	2,7	вересень

Створення і запис текстового файлу. Відкриття файлу, позиціонування у файлі, читання файлу.			(4 год.)		
Тема 7. <b>Керування пам'яттю і відображенням файлів</b> Керування пам'яттю купи. Відображення файлів.	лекція	6, с. 155–185	Опрацювати літературу, прочитати конспект (4 год.)	0,3	вересень
Тема 8. <b>Читання і запис файлів засобами Windows API (частина 2)</b> Складання програми для читання і редагування текстового файлу. Визначення розміру файлу.	лабораторне заняття	1, 2, 6	Виконати лабораторну роботу (4 год.)	2,7	вересень
Тема 9. <b>Керування процесами (базові можливості)</b> Читання командного рядка. Процеси і потоки в Windows. Створення процесу. Завершення процесу. Очікування завершення процесу.	лекція	3, 6	Опрацювати літературу, прочитати конспект (4 год.)	0,3	жовтень
Тема 10. <b>Керування файлами і каталогами засобами Windows API (частина 1)</b> Створення каталогу. Копіювання файлів. Зміна атрибутів файлу. Блокування і розблокування файлу.	лабораторне заняття	1, 2, 6	Виконати лабораторну роботу (4 год.)	2,7	жовтень
Тема 11. <b>Керування процесами (додаткові можливості)</b> Ідентифікатори процесів. Спадкування і дублювання дескрипторів. Блоки і рядки оточення. Приклади.	лекція	3, 6	Опрацювати літературу, прочитати конспект (4 год.)	0,3	жовтень
Тема 12. <b>Керування файлами і каталогами засобами Windows API (частина 2)</b> Читання вмісту каталогу. Перейменування файлів. Знищення файлів.	лабораторне заняття	1, 2, 6	Виконати лабораторну роботу (4 год.)	2,7	жовтень
Тема 13. <b>Потоки і планування виконання</b> Переваги і недоліки багатопотокових процесів. Керування потоками. Очікування завершення потоків і віддалені потоки. Пріоритети процесів і потоків і планування виконання. Стани потоків. Приклад.	лекція	3, с. 55–73, 6, с. 207–264	Опрацювати літературу, прочитати конспект (4 год.)	0,3	жовтень

Тема 14. <b>Редагування файлів з допомогою їхнього відображення в пам'ять засобами Windows API</b> Створення програми для відображення файлу в пам'ять і його шифрування.	лабораторне заняття	1, 2, 3, 6	Виконати лабораторну роботу (4 год.)	2,7	жовтень
Тема 15. <b>Синхронізація потоків (частина 1)</b> Необхідність синхронізації потоків і правила безпечного багатопотокового коду. Функції взаємоблокування. Об'єкти синхронізації потоків. Критичні секції. Приклад.	лекція	3, с. 74–78, 6, с. 271–285	Опрацювати літературу, прочитати конспект (4 год.)	0,3	жовтень
Тема 16. <b>Керування процесами (частина 1)</b> Створення програми для виконання певних дій.	лабораторне заняття	1, 2, 3, 6	Виконати лабораторну роботу (4 год.)	2,7	жовтень
Тема 17. <b>Синхронізація потоків (частина 2)</b> М'ютекси. Семафори. Події. Приклад.	лекція	3, с. 79–87, 6, с. 286–306	Опрацювати літературу, прочитати конспект (4 год.)	0,3	листопад
Тема 18. <b>Керування процесами (частина 2)</b> Запуск програми як процесу, передача процесу даних і очікування його завершення.	лабораторне заняття	1, 2, 3, 6	Виконати лабораторну роботу (4 год.)	2,7	листопад
Тема 19. <b>Взаємодія між процесами (частина 1)</b> Засоби взаємодії між процесами. Анонімні канали. Приклад створення анонімного каналу.	лекція	3, 6	Опрацювати літературу, прочитати конспект (4 год.)	0,3	листопад
Тема 20. <b>Створення багатопотокових програм</b> Створення програми для виконання певних дій, в якій батьківський потік запустить дочірній потік, передасть йому вхідні дані, дочекається його завершення і виведе результати його роботи.	лабораторне заняття	1, 2, 3, 6	Виконати лабораторну роботу (4 год.)	2,7	листопад
Тема 21. <b>Взаємодія між процесами (частина 2)</b> Іменовані канали. Підключення іменованих каналів. Функції транзакцій іменованих каналів. Поштові скриньки.	лекція	3, 6	Опрацювати літературу, прочитати конспект (4 год.)	0,3	листопад
Тема 22. <b>Керування</b>	лабора-	1, 2, 3, 6	Виконати	2,7	листопад

<b>пріоритетами потоків</b> Запуск двох паралельних потоків з заданими пріоритетами, передача їм вхідних даних, очікування завершення їхньої роботи, виведення результатів і часу роботи цих потоків.	торне заняття		лабораторну роботу (4 год.)		
<b>Тема 23. Синхронізація потоків</b> Розв'язання заданої задачі про обід філософів шляхом запуску кількох паралельних потоків і синхронізації їхньої роботи з допомогою критичних секцій, м'ютексів, семафорів або об'єктів події.	лабораторне заняття	1, 2, 3, 6	Виконати лабораторну роботу (4 год.)	2,7	листопад
<b>Тема 24. Основи асемблера</b> Асемблер і системне програмування. Регістри процесора. Етапи створення програми і структура програми на асемблері. Директиви асемблера. Приклад програми на асемблері.	лекція	4, 5	Опрацювати літературу, прочитати конспект (4 год.)	0,3	грудень
<b>Тема 25. Передача даних між процесами з допомогою іменованого каналу</b> Створення процесу і іменованого каналу. Обмін даними між процесами через канал.	лабораторне заняття	1, 2, 3, 6	Виконати лабораторну роботу (4 год.)	2,7	грудень
<b>Тема 26. Основні конструкції асемблера</b> Команди передачі даних. Арифметичні операції. Цикл. Умовний і безумовний переходи. Стек і підпрограми. Конструкції часу виконання програми. Макроси. Приклад консольної програми.	лекція	4, 5	Опрацювати літературу, прочитати конспект (4 год.)	0,3	грудень
<b>Тема 27. Приклади програмування на асемблері</b>	лекція	4, 5	Опрацювати літературу, прочитати конспект (4 год.)	0,3	грудень
<b>Тема 28. Знайомство з асемблером</b> Створення програми на асемблері.	лабораторне заняття	4, 5	Виконати лабораторну роботу (4 год.)	2,7	грудень
<b>Тема 29. Контрольна робота</b> Створення програми для	лабораторне заняття	1, 2, 3, 6	Підготуватись до контрольної роботи	10	грудень

виконання обчислень і передачі даних між потоками або процесами.			(4 год.)		
Тема 30. Підсумкове заняття Повторення вивченого матеріалу. Захист раніше не захищених лабораторних робіт.	лабораторне заняття	1 – 6	Повторити вивчений матеріал (4 год.)	0,3	грудень

#### 7. Система оцінювання курсу

Загальна система оцінювання курсу	Система оцінювання курсу здійснюється згідно з критеріями оцінювання навчальних досягнень студентів, що регламентовані в університеті. Допуск до іспиту становить максимум 50 балів, бал за складання іспиту (підсумковий контроль) становить максимум 50 балів.
Вимоги до письмової роботи	Передбачено одну контрольну роботу, яка оцінюється за шкалою від 0 до 10 балів.
Практичні заняття	Оцінюється відвідуваність усіх занять упродовж семестру за 10-бальною шкалою. Оцінюється виконання лабораторних робіт за 30-бальною шкалою.
Умови допуску до підсумкового контролю	При виставленні допуску до іспиту (максимум 50 балів) враховуються навчальні досягнення студентів (бали), набрані під час контактних (аудиторних) годин, при виконанні завдань для самостійної роботи, а також бали за контрольну роботу.

#### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	<b>A</b>	відмінно
80 – 89	<b>B</b>	добре
70 – 79	<b>C</b>	
60 – 69	<b>D</b>	задовільно
50 – 59	<b>E</b>	
25 – 49	<b>FX</b>	незадовільно
0 – 24	<b>F</b>	

#### 8. Політика курсу

Загальна максимальна сума балів, яка присвоюється студентові за вивчення навчальної дисципліни, становить 100 балів – сума балів за виконання лабораторних робіт, контрольної роботи, відвідування навчальних занять та бали, отримані під час іспиту. Допуск до іспиту передбачає отримання рейтингової підсумкової оцінки (максимум 50 балів, мінімум 25 балів).

При виставленні рейтингового підсумкового балу обов'язково враховується присутність студента на заняттях (у тому числі на лекційних), активність студента під час практичних занять, наявність пропусків без поважних причин, користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час опитування та виконання письмових завдань, списування та плагіат, а також результати відпрацювання пропущених з поважної причини занять.

Студент, який не набрав 25 балів, до іспиту за відомістю № 1 не допускається. У такому випадку до початку екзаменаційної сесії або під час ліквідації академічної заборгованості студент користується повторним правом отримати допуск на складання іспиту за відомістю № 2 на консультаціях викладача (перескладання пропущених тем, виконання індивідуальних завдань і контрольних робіт).

#### 9. Рекомендована література



1. Глинський Я. М., Анохін В. Є., Рязька В. А. С++ і С++ Builder. Львів : СПД Глинський, 2011.
2. Щеряков С.М. Основи мови асемблер. Івано-Франківськ, 2004. 52 с.
3. Коноваленко І. В., Федорів П. С. Системне програмування у Windows з прикладами на Delphi. Тернопіль : ТНТУ ім. І. Пулюя, 2012. 320 с.
4. Fog A. Optimizing subroutines in assembly language. An optimization guide for x86 platforms. Lyngby: Technical University of Denmark, 2019. 169 p.
5. Hart J.M. Windows System Programming. Addison-Wesley, 2010. 645 p.
6. Kusswurm D. Modern X86 Assembly Language Programming: Covers X86 64-bit, AVX, AVX2, and AVX-512. Apress, 2023. 688 p.
7. Yosifovich Pavel. Windows 10 System Programming, Part 1, 2. Lean Publishing, 2020, 2021. 620, 625 p.

**Викладач Махней Олександр Володимирович**