

# Об'єктно-орієнтоване програмування (ознайомча практика)

## СИЛАБУС

<b>Шифр і назва спеціальності</b>	122 – комп'ютерні науки	<b>Інститут / факультет</b>	Комп'ютерних наук і програмної інженерії
<b>Назва програми</b>	«Комп'ютерні науки та інтелектуальні системи»	<b>Кафедра</b>	Програмної інженерії та інформаційних технологій управління
<b>Тип програми</b>	Освітньо-професійна	<b>Мова навчання</b>	українська

### Викладач

**Нікуліна Олена Миколаївна**

Olena.Nikulina@khpi.edu.eu



д.т.н. доцент, професор кафедри програмної інженерії та інформаційних технологій управління. Підготував і опублікував понад 90 публікацій (h-index = 5, i10-index = 2 in Google Scholar - [https://scholar.google.com/citations?hl=en&user=ZEe2GlcAAAAJ&view\\_op=list\\_works&sortby=title](https://scholar.google.com/citations?hl=en&user=ZEe2GlcAAAAJ&view_op=list_works&sortby=title); ідентифікатор ORCID-<https://orcid.org/0000-0003-2938-4215>, ідентифікатор автора Scopus-57203114988).

**Провідний лектор з курсів:** *Об'єктно-орієнтоване програмування (бакалаври) (українською мовою), Чисельні методи (бакалаври) (українською мовою), Дослідження операцій (бакалаври) (українською мовою), Інтелектуальні системи управління (бакалаври) (українською мовою), Моделі та програмні засоби розподілених обчислень (PhD) (українською мовою)*

### Загальна інформація про курс

<b>Анотація</b>	Курс «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ» є навчальною дисципліною з циклу професійної обов'язкової підготовки за спеціальністю 122 «комп'ютерні науки». Вона викладається у четвертому семестрі в обсязі 120 год.(4 кредитів ECTS), зокрема: лекції – 32 год., лабораторні – 32 год., самостійна робота – 56 год. У курсі передбачено три змістових модулі та три контрольні роботи. Завершується дисципліна іспитом.  Дисципліна взаємопов'язана з такими дисциплінами як «Вступ до фаху» та «Алгоритмізація та програмування»
<b>Цілі курсу</b>	Вивчити основи проектування програмного забезпечення; вивчити технології об'єктно-орієнтованого програмування; вивчити прийоми роботи з візуальними середовищами програмування; набуття навичок розробки й тестування програмних продуктів функціонуючих під керуванням сучасних операційних систем; формування у студентів абстрактного мислення, яке повинне допомогти рішенню прикладних задач, пов'язаних з різноманітними галузями знань. Засвоєння необхідних знань з опанування сучасними технологіями об'єктно-орієнтованого аналізу, проектування та програмування об'єктно-орієнтованої моделі різними мовами програмування
<b>Формат</b>	Лекції, лабораторні роботи, самостійна робота, консультації, курсова робота. Підсумковий контроль – іспит.
<b>Семестр</b>	3

**Обсяг (кредити) / Тип курсу**  
(обов'язковий / вибіркового)

4 / Обов'язковий

**Лекції (години)**

32

**Лабораторні  
заняття (години)**

32

**Самостійна робота (години)**

56

<b>Програмні компетентності</b>	<p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК6. Здатність вчитися й оволодівати сучасними знаннями.</p> <p>ЗК9. Здатність працювати в команді.</p> <p>СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмів управління.</p>
---------------------------------	---

Результати навчання	Методи викладання та навчання	Форми оцінювання (поточне оцінювання CAS, підсумкове оцінювання FAS)
<p>ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проектне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), експрес-опитування (CAS), підсумковий/семестровий контроль у формі семестрового іспиту, відповідно до графіку навчального процесу (FAS)</p>
<p>ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проектне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), збір даних за індивідуальними завданнями та звітування за результатами дослідження (CAS), підсумковий/семестровий контроль у формі семестрового іспиту, відповідно до графіку навчального процесу (FAS)</p>
<p>ПР14. Застосовувати знання методології та CASE-засобів проєктування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проєктування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, дослідження, проектне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), збір даних за індивідуальними завданнями та звітування за результатами дослідження (CAS), підсумковий/семестровий контроль у формі семестрового іспиту, відповідно до графіку навчального процесу (FAS)</p>

### СИСТЕМА ОЦІНЮВАННЯ

балів для оцінювання успішності студента	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів	100% підсумкове оцінювання у вигляді іспиту (20%) та поточного оцінювання (80%). 20% іспит: семестровий іспит, відповідно до графіку навчального
	90-100	A	відмінно		
	82-89	B	добре		
	74-81	C			

	64-73	D	задовільно		процесу <b>80% поточне оцінювання:</b> <ul style="list-style-type: none"> <li>• 30% оцінювання завдань на лабораторних роботах;</li> <li>• 30% проміжний контроль (3 контрольні роботи)</li> <li>• 20% курсова робота</li> </ul>
	60-63	E			
	35-59	FX	незадовільно з можливістю повторного складання		
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни		

**Політика курсу** Дотримуватися правил внутрішнього розпорядку університету. Брати активну участь у навчальному процесі. Студенти зобов'язані відвідувати заняття згідно розкладу та дотримуватися етики поведінки. У разі відсутності студентам необхідно буде виконати всі завдання, щоб компенсувати пропущені заняття. Виконання лабораторних робіт вимагає попередньої підготовки та завчасного опрацювання всіх необхідних матеріалів. Письмові завдання повинні бути подані до встановлених строків.

### Структура та зміст курсу

<b>Лекція 1</b>	Створення та використання класів C++. Інкапсуляція. Обробка винятків	<b>Лабораторна робота 1</b>	Класи та об'єкти у C++. Перевантаження операцій Innovation Campus: ПЗ08	<b>Самостійна робота</b>	
<b>Лекція 2</b>	Успадкування. Використання поліморфізму та шаблонів у C++	<b>Лабораторна робота 2</b>	Композиція класів. Успадкування. Innovation Campus: ПЗ08, ПЗ10		
<b>Лекція 3</b>	Використання засобів стандартної бібліотеки C++.	<b>Лабораторна робота 3</b>	Віртуальні та абстрактні класи. Винятки. Innovation Campus: ПЗ08, ПЗ10, ПЗ11		
<b>Лекція 4</b>	Контейнери C++. Алгоритми C++.	<b>Лабораторна робота 4</b>	Контейнери. Шаблони. стандартні бібліотеки. Innovation Campus: ПЗ08, ПЗ10		
<b>Лекція 5</b>	Використання базових засобів мови Java	<b>Лабораторна робота 5</b>	Графічний інтерфейс користувача Innovation Campus: ПЗ11		
<b>Лекція 6</b>	Робота з масивами та рядками Java. Створення класів	<b>Лабораторна робота 6</b>	Класи та об'єкти у Java. Вкладені класи. Композиція. Innovation Campus: ПЗ08, ПЗ21		
<b>Лекція 7</b>	Використання поліморфізму. Робота з узагальненнями та колекціями в Java	<b>Лабораторна робота 7</b>	Успадкування. інтерфейси та абстрактні класи. Innovation Campus: ПЗ21		
<b>Лекція 8</b>	Робота з винятками і файлами в Java	<b>Лабораторна робота 8</b>	Винятки. Узагальнення. Innovation Campus: ПЗ21		Робота з потоками символів і потоками байтів

<b>Лекція 9</b>	Робота з XML-документами в Java	<b>Лабораторна робота 9</b>	Контейнери. Innovation Campus: ПЗ22	
<b>Лекція 10</b>	Створення застосунку графічного інтерфейсу користувача Java	<b>Лабораторна робота 10</b>	Графічний інтерфейс користувача Innovation Campus: ПЗ22	Архітектура JavaFX-застосунку.
<b>Лекція 11</b>	Використання базових засобів мови C#.	<b>Лабораторна робота 11</b>	Класи та об'єкти у C#. Перевантаження операцій Innovation Campus: ПЗ23	
<b>Лекція 12</b>	Робота з масивами та рядками C#. Створення класів. Успадкування.	<b>Лабораторна робота 12</b>	Вкладені класи. Композиція. Успадкування. Innovation Campus: ПЗ23	Структури і перелічення
<b>Лекція 13</b>	Поліморфізм C#. Інтерфейси. Проектування та реалізація узагальнених класів і методів Обробка виняткових ситуацій.	<b>Лабораторна робота 13</b>	Віртуальні та абстрактні класи. Винятки. Innovation Campus: ПЗ23	
<b>Лекція 14</b>	Роботу з файлами. Робота з XML-документами C#. Створення та використання узагальнень.	<b>Лабораторна робота 14</b>	Узагальнення. Контейнери. Innovation Campus: ПЗ24	Створення власних контейнерних типів.
<b>Лекція 15</b>	Створення .NET-застосунку графічного інтерфейсу користувача C# Делегати. Події.	<b>Лабораторна робота 15</b>	Графічний інтерфейс користувача Innovation Campus: ПЗ24	Графічні засоби GDI+
<b>Лекція 16</b>	Використання уніфікованої мови моделювання (UML) Основи опису та використання патернів проектування.	<b>Лабораторна робота 16</b>	UML діаграми Innovation Campus: ПЗ10, ПЗ24	

### Література

1. Bjarne Stroustrup. (2007). The C++ Programming Language. (3rd Ed.). Addison-Wesley,.
2. Stanley, B. Lippman, Josee Lajoie. (2008). C++ Primer. (3rd Ed.). Addison-Wesley.
3. Савич, У. (2004). Програмування на C++. Київ: BHV.
4. Нікуліна, О. М. (2014). Основи програмування у візуальному середовищі. Методичні вказівки до лабораторних занять з курсу «Системне програмування». Харків: НТУ «ХПІ».
5. Deitel, H. M., Deitel, P. J. (2001). C++. How to Program. (3rd Ed.). Prentice Hall.
6. Грицюк, Ю. І., Рак Т. Є. (2011). Програмування мовою C++: Навч. посіб. Львів: Вид-во Львівського ДУ БЖД.
7. Shieldt, G., Holmes, D. (2005). The art of Java programming. Publishing house «Williams».
8. Horstmann Ks, Cornell G. (2007). Java 2. Library of the Professional. (Vol. 1). (7th ed.). Publishing house «Williams».
9. Копитко, М. Ф., Іванків, К. С. (2002). Основи програмування мовою Java: Тексти лекцій. Львів: Видав. центр ЛНУ ім. Івана Франка.
10. Голуб, Б. М. (2006). C#. Концепція та синтаксис: Навч. посіб. Львів: Видав. центр ЛНУ ім. Івана Франка.
11. Брнакевич, І. Є., Вагін, П. П. (2002). Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. Львів: Видав. центр ЛНУ ім. Івана Франка.
12. Deitel, H. M., Deitel, P. J. (2018). Java. How to Program. (10th ed.). Prentice Hall.

1. Walpa, O. D., Borland, (2016). C++ Builder. Express course.
2. Іванов, Л. В. Основи програмування (Ч. 2). Retrieved from [http://www.iwanoff.inf.ua/programming\\_2\\_ua/index.html](http://www.iwanoff.inf.ua/programming_2_ua/index.html)
3. Deytel, H. M. (2016). Deytel How to Program in Java. (Book 2). Files, networks, databases.
4. Іванов, Л. В. Об'єктно-орієнтоване програмування (Ч. 1). Retrieved from [http://www.iwanoff.inf.ua/oop\\_ua/index.html](http://www.iwanoff.inf.ua/oop_ua/index.html)
5. Richter, J. (2017). Programming on Microsoft. NET Framework 2.0 in C #.
6. Richter, J., Van de Bospurt, M. (2014). Programming in C # for professionals. Williams.
7. Іванов, Л. В. Об'єктно-орієнтоване програмування (Ч. 2). Retrieved from [http://www.iwanoff.inf.ua/oop\\_ua/index.html](http://www.iwanoff.inf.ua/oop_ua/index.html)
8. Design Patterns: Elements of Reusable Object-Oriented Software. <http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>
9. Брнакевич, І. Є., Вагін, П. П. (2002). Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. Львів: Видав. центр ЛНУ ім. Івана Франка.
10. Дудзяний, І. М. (2007). Об'єктно-орієнтоване моделювання програмних систем: Навч. посіб. Львів: Видав. центр ЛНУ ім. Івана Франка.

### Норми академічної етики

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність

Силабус за змістом повністю відповідає робочій програмі курсу.