

Порівняння дисциплін

для формування програми міжнародної мобільності в рамках програми Erasmus+ K107
на основі [переліку дисциплін](#), що викладаються англійською мовою в UPV у 2020/2021 н.р.

Таблиця 1. Навчальний план для студентів 4 курсу (осінній семестр) спеціальності 121 «Інженерія програмного забезпечення» на 2021-2022 н.р. та дисципліни-аналоги Політехнічного університету Валенсії

НТУ, ПЗ, освітній рівень - бакалавр				UPV, School of Informatics, <u>Bachelor's Degree in Informatics Engineering</u>			
Шифр	Назва освітньої компоненти	Кількість кредитів	Семестр(-и)/ вид контролю	Шифр	Назва освітньої компоненти	Кількість кредитів	Семестр
ОКП12	Емпіричні методи програмної інженерії	3	7 залік	11555	Software Engineering [Compulsory]	6	Semester A (Sept/Jan)
ОКП20	Моделювання та аналіз програмного забезпечення	3	7 екзамен				
ОКП16	Якість програмного забезпечення та тестування	4	7 залік	11566	Software analysis, validation and debugging [Elective]	4.5	Semester A (Sept/Jan)
ОКП17	Організація комп'ютерних мереж	(3+4)	7-8 екзамен (7) залік+КР (8)	12990	Computer Networks [Compulsory]	9 (2*4.5)	Semester A + B (Sept/July)
ОКП25	Інтелектуальний аналіз даних	4	7 екзамен+КР				
ОКП27	Методи та системи штучного інтелекту	(2+3)	7-8 залік (7) екз. + КР(8)	11560	Intelligent Systems [Compulsory]	4,5	Semester A (Sept/Jan)
ВБК 10.1	Теорія розпізнавання образів	4	7 залік	11557	Programming Languages, Technologies, And Paradigms [Compulsory] / або іншу, запроповану UPV в переліку дисциплін	6	Semester A (Sept/Jan)
/ВБК 10.2	/Мікропроцесорні системи на транспорті	/4	7 залік				
ВБК 14.1	Технології розподілених систем та паралельних обчислень	4	7 Іспит+КР				
/ВБК 14.2	/Крос-платформне (Прикладне) програмування	/4	7				
		30				25.5	

Практика ----

«/» - «або»

**Таблиця 1. Порівняння дисциплін для забезпечення процесу перезарахування освітніх компонент освітньої програми спеціальності 121
«Інженерія програмного забезпечення»**

НТУ, ІСТ					
Дисципліна	Структура дисципліни	Пре-реквізити			Prior Knowledge
<p>Емпіричні методи програмної інженерії</p> <p>Івохін Євген Вікторович</p>	<p><i>ЗМ 1. Методи розробки програмних продуктів і систем</i></p> <p>1. Процеси, продукти, ресурси.</p> <p>2. Складові життєвого циклу програмного забезпечення.</p> <p><i>ЗМ 2. Класифікація методів емпіричних засобів</i></p> <p>3. Основи інженерії програмного забезпечення.</p> <p>4. Основні інструменти інженерії ПЗ.</p> <p><i>Змістовний модуль 3. Емпіричні методи інженерії програмного забезпечення</i></p> <p>5. Технології емпіричної інженерії ПЗ.</p> <p>6. Взаємозв'язок інженерій.</p> <p><i>ЗМ 4. Моделювання життєвого циклу програмного забезпечення</i></p> <p>7. Типи моделей життєвого циклу.</p> <p>8. Вибір моделей життєвого циклу.</p> <p>Основи процесів тестування.</p>	<p>Об'єктноорієнтоване програмування,</p> <p>Теорія ймовірності та мат. статистика,</p> <p>Безпека програм та даних</p>			
<p>Якість програмного забезпечення та тестування</p> <p>Сватко Віталій Володимирович</p>	<p><i>ЗМ 1. Тестування як засіб підвищення надійності програмного забезпечення</i></p> <p>1. Основні визначення. Рівні і види тестування. Техніка тестування. Аксиоми тестування.</p> <p>2. Основи якості програмного забезпечення. Процеси управління якістю програмного забезпечення.</p> <p><i>ЗМ 2. Принципи тестування програмного забезпечення</i></p>	<p>Конструювання ПЗ,</p> <p>Основи програмної інженерії,</p> <p>Алгоритми та структури даних</p>	<p>Software analysis, validation and debugging</p>	<p>1. <i>Introduction</i></p> <p>2. <i>Validation</i></p> <p>2.1. Software testing</p> <p>2.2. Symbolic execution</p> <p>2.3. Automatic generation of test cases</p> <p>2.4. Progress and challenges</p> <p>3. <i>Analysis</i></p> <p>3.1. Data flow analysis</p> <p>3.1. Interprocedural analysis</p> <p>4. <i>Debugging</i></p> <p>4.1. Debugging</p> <p>4.2. Slicing</p>	<p>There is not identified Recommended Prior Knowledge</p>

	<p>3. Основні принципи та підходи до тестування програмного забезпечення. Процес тестування ПЗ.</p> <p>4. Випадкове (стохастичне) тестування. Тестування за класами еквівалентності. Метод аналізу граничних умов. <i>ЗМ 3. Інтеграційне тестування компонентно-базового програмного забезпечення</i></p> <p>5. Тестування потоків керування програми. Тестування потоків даних програми. Мутаційне тестування.</p> <p>6. Критерії й метрики інтеграційного тестування. Ієрархія й відповідність між критеріями інтеграційного тестування. <i>ЗМ 4. Оцінювання кількості тестів на всіх етапах життєвого циклу програмного забезпечення</i></p> <p>7. Оцінювання кількості тестів для інтеграційного тестування. Створення тестів на основі UML діаграм варіантів використання.</p> <p>8. Оцінювання часу тестування за допомогою UML діаграм варіантів використання.</p>			<p>This course offers a general perspective, detached from any particular language or programming style, of techniques and tools for the analysis, validation and debugging of software in order to meet the requirements of reliability and robustness demanded by the user, standards and society. After the general vision, the subject presents in concrete the most extended and effective techniques at a practical level.</p> <p>The course is taught in reverse mode, so the student will have at his disposal videos and other study material (in polyformat) to work on before the theory sessions. The face-to-face sessions will consist mainly of the performance of tasks (individual or in group), resolution of doubts, etc. The morning group will be taught primarily in English. All students will have the material in both Spanish and English.</p> <hr/> <p>References</p> <p>Principles of model checking (Baier, Christel Katoen, Joost-Pieter)</p> <p>Principles of program analysis (Nielson, Flemming Hankin, Chris Nielson, Hanne Riis)</p> <p>Software reliability methods (Peled, Doron A)</p> <p>A vocabulary of program slicing-based techniques (Silva, Josep)</p> <p>Test Generation Using Symbolic Execution (Godefroid, Patrice)</p> <hr/> <p>Software Quality (Spanish)</p> <p>Syllabus</p> <p>1. Topic 1: Fundamentals of Software Quality</p> <p>2. Topic 2: Quality Perspectives</p> <p>1. Product Quality: Standards, Quality Models</p> <p>2. Quality of Software Architectures</p>	
--	---	--	--	---	--

				<ul style="list-style-type: none"> 3. Process Quality: Standards, Evaluation Models and Process Improvement 4. Quality in Use: Standards, Usability Evaluation Methods 3. Topic 3: Software Metrics <ul style="list-style-type: none"> 1. The Objective/Question/Metrics (GQM) Paradigm 2. Definition of Quality Metrics and Indicators 3. Theoretical and Empirical Validation 4. Item 4: Quality Management, Assurance and Control <ul style="list-style-type: none"> 1. Software Quality Management 2. Software Quality Assurance: activities and methods 3. Quality Control 5. Item 5: Estimation of Software Projects <ul style="list-style-type: none"> 1. Functional Size Measurement Methods 2. Definition of Indicators (productivity, effort, cost, etc.) 6. Internships 	
Організація комп'ютерних мереж Зайцев Євген Олександрович	<i>ЗМ 1. Основи комп'ютерних мереж.</i> Тема 1. Загальні принципи побудови комп'ютерних мереж (КМ) Тема 2. Основні принципи організації взаємодії в КМ <i>ЗМ 2. Апаратно-програмні засоби комп'ютерних мереж.</i> <i>Архітектура мереж.</i> Тема 3. Апаратні засоби КМ та їх функції.	Архітектура комп'ютера, Основи програмування, Бази даних, Операційні системи, Іноземна мова,	Computer Networks <u>1</u>	<ul style="list-style-type: none"> 1. TCP/IP communications architecture <ul style="list-style-type: none"> 1.1. Introduction to computer networks and the Internet 1.2. Network applications 1.3. Network programming interface: the sockets 1.4. The transport level 2. Security in communications <ul style="list-style-type: none"> 2.1. Computer Network Security 3. Data communications <ul style="list-style-type: none"> 3.1. The network level 	There is not identified Recommended Prior Knowledge

	<p>Тема 4. Локальні комп'ютерні мережі. 8 семестр: ЗМ 3. Модель OSI Тема 5. Взаємодія рівнів еталонної моделі OSI. Тема 6. Рівні моделі OSI ЗМ 4. Комп'ютерні мережі та мережеві технології. Тема 7. Глобальні комп'ютерні мережі та мережеві технології. Тема 8. Програмне забезпечення комп'ютерних мереж. <i>Лабораторні роботи:</i> Тема 1. Загальні принципи побудови комп'ютерних мереж Тема 2. Основні принципи організації взаємодії в КМ Тема 3. Апаратні засоби КМ та їх функції. Тема 4. Локальні комп'ютерні мережі. Тема 5. Взаємодія рівнів еталонної моделі OSI. Тема 6. Рівні моделі OSI. Тема 7. Глобальні комп'ютерні мережі та мережеві технології. Тема 8. Програмне забезпечення комп'ютерних мереж.</p>	<p>Українська мова (за професійним спрямуванням)</p>		<p>3.2. The level of data link and local area networks 3.3. The physical level</p> <p>The course introduces the student to the study of communications between computer systems, based on the TCP/IP architecture currently used on the Internet. The approach used is descending, beginning with the study of the level of application, the last level of architecture but the most familiar to the student, and continuing with the lower levels.</p> <p>Since this is the first subject on communications in the curriculum, it has a general and descriptive character, dealing with the different aspects of computer networks but without delving into the more complex theoretical issues.</p>	
<p>Моделювання та аналіз програмного забезпечення</p>	<p><i>ЗМ 1. Моделювання програмного забезпечення.</i> 1. Основні принципи моделювання (декомпозиція, абстракція, узагальнення, демонстрація, використання формальних підходів). 2. Передумови, постумови, інваріанти та обумовлене проектування (за контрактом).</p>	<p>Архітектура та проектування програмного забезпечення, Аналіз вимог до програмного забезпечення</p>	<p>Software Engineering</p>	<p>1. Introduction to Software Engineering 2. The Software Process 3. Software Architecture 1. 3-Layer Architecture 4. Object Oriented Modeling 2. Use Cases in UML 3. Diagram of Classes in UML 4. Other diagrams</p>	<p>Programming</p>

	<p>3. Вступ до математичного моделювання та формальні нотації <i>ЗМ 2. Види моделей</i></p> <p>4. Інформаційне моделювання.</p> <p>5. Моделювання поведінки</p> <p>6. Архітектурне моделювання.</p> <p>7. Моделювання предметної області.</p> <p><i>ЗМ 3. Аналіз програмного забезпечення.</i></p> <p>8. Основи аналізу програмного забезпечення.</p> <p>9. Аналіз форми (завершеність, узгодженість даних, надійність)</p> <p>10. Формальний аналіз коду.</p>			<p>5. Software Design</p> <p>5. Application Logic Design</p> <p>6. Persistence Design</p> <p>7. User Interface Design</p> <p>6. Tests</p> <p>The aim of the course is to present current methods, techniques and tools for the development of quality software. For this purpose, the object-oriented paradigm will be used throughout the entire development life cycle. As this is the first subject related to the discipline of software engineering, we will focus mainly on development aspects, leaving the management aspects for subjects of the corresponding branch.</p> <p>The theory classes present the basic principles of object-oriented software development, in particular multi-layer architecture, modeling, design, implementation and testing. In the seminar classes the acquired knowledge is reinforced with problems and exercises, many of them related to the project of the subject. The development of a complete software project will be the main objective of the laboratory practice classes, in which group work will be carried out.</p>	
<p>Інтелектуальний аналіз даних</p> <p>Зубрецька Н.А.</p>	<p><i>ЗМ 1. Технології аналізу та трансформації даних методами Data Mining</i></p> <p>1. Технології аналізу даних</p> <p>2. Трансформація даних</p> <p><i>ЗМ 2. Попередня обробка та оцінювання якості даних</i></p> <p>3. Візуалізація даних</p> <p>4. Оцінка якості та попередня обробка даних</p> <p><i>ЗМ 3. Асоціативні правила та методи кластеризації даних</i></p> <p>5. Пошук асоціативних правил</p> <p>6. Кластеризація</p> <p><i>ЗМ 4. Часові ряди та машинне навчання з теми</i></p>	<p>Теорія ймовірностей, ймовірнісні процеси та математична статистика</p> <p>Бази даних/Системи управління базами даних</p> <p>Методи та системи штучного інтелекту</p>			

	7. Аналіз та прогнозування часових рядів 8. Класифікація і регресія, машинне навчання				
<p>Методи та системи штучного інтелекту</p> <p>Зубрецька Н.А.</p>	<p>Модуль 1. Основи інтелектуальних систем <i>ЗМ 1. Основні поняття та визначення СШ</i></p> <p>1. Основи систем штучного інтелекту 2. Історія методів і систем штучного інтелекту <i>ЗМ 2. Машинне навчання</i></p> <p>3. Формалізація інтелектуальної задачі 4. Пошук рішень інтелектуальної задачі</p> <p>Модуль 2. Представлення знань у СШ <i>ЗМ 3. Проблеми представлення знань</i></p> <p>5. Знання та моделі представлення знань у СШ 6. Продукційні моделі <i>ЗМ 4. Методи представлення знань</i></p> <p>7. Класифікація систем, які засновані на знаннях 8. Основи експертних систем</p> <p>Семестр 8:</p> <p>Модуль 3. Розподілені інтелектуальні системи Змістовний модуль 5. Інтелектуальні системи планування та прогнозування 9. Нейромережне моделювання</p> <p>10. Нейронне програмування діалогових систем Змістовний модуль 6. Основи мультиагентних систем</p>	<p>Теорія ймовірностей, ймовірнісні процеси та математична статистика</p> <p>Бази даних/Системи управління базами даних</p> <p>Інтелектуальний аналіз даних</p>	<p>Intelligent Systems</p>	<p>1. Presentation of the subject 2. Intelligent Systems: Representation and Search</p> <p>2.1 Introduction to AI. Concepts, evolution, areas and applications. 2.2. Rule-based systems (SBR). Representation in SBR: facts and rules. Pattern-matching. 2.3. Inference in SBR: chaining and control, RETE. 2.4. Design of problems based on SBR states. 2.5. Problem solving based on states: unreported search. 2.6. Problem solving based on states: heuristic search, A* algorithm. 2.7. Search among adversaries. Minimax Algorithm, Alpha-Beta.</p> <p>3. Intelligent Systems: Automatic Learning</p> <p>2.8. Introduction to Form Recognition and Automatic Learning. Probabilistic reasoning. 2.9. Learning discriminant functions. Perceptron. 2.10. Induction of rules and patterns. Decision trees. 2.11. Non-supervised learning: k-medias algorithm. 2.12. Structured representation. Markov models. Forward algorithm. 2.13. Viterbi's algorithm. Estimation of Markov models. Re-estimation by Viterbi.</p> <p>- Introduction to Intelligent Systems. Concepts, evolution, areas and applications. - Artificial Intelligence problem solving. Heuristic search. Search with adversary. - Representation of knowledge and</p>	<p>There is not identified Recommended Prior Knowledge</p>

	<p>11. Архітектура мультиагентних систем</p> <p>12. Проектування мультиагентних систем</p> <p>Модуль 4. Нечітке моделювання та управління</p> <p>Змістовний модуль 7. Основи нечіткої логіки</p> <p>13. Моделі знань на основі нечіткої логіки</p> <p>Тема нечіткої логіки у системах управління</p> <p>Змістовний модуль 8.</p> <p>Розроблення інтелектуальних систем на основі нечіткої логіки</p> <p>15. Створення нечіткої логічної системи управління</p> <p>16. Практична реалізація нечітких систем управління</p>			<p>inference. Rule-based systems. Chaining and control. Probabilistic Reasoning. - Automatic learning. Learning discriminant functions, decision trees. Unsupervised learning. Markov models, Viterbi algorithm.</p>	
Вибіркові освітні компоненти					
<p>Теорія розпізнавання образів</p> <p>Ковальчук Оксана Петрівна</p>	<p><i>ЗМ 1. Підхід в теорії розпізнавання образів. Класифікація систем розпізнавання. Класи та їх властивості.</i></p> <p>1. Предмет розпізнавання образів. Основні задачі теорії розпізнавання образів. Типи характеристик образів. Типи систем розпізнавання. Математична постановка задач розпізнавання. Попередня обробка сигналів та зображень.</p> <p>2. Поняття вирішальних функцій. Лінійні вирішальні функції (ЛВФ). Загальний підхід до знаходження лінійних вирішальних функцій. Алгоритм ХоКашьяпа. Узагальнені вирішальні функції. Задача пониження розмірності.</p>	<p>Методи та системи штучного інтелекту</p> <p>Інформаційні системи на транспорті</p> <p>Основи проектування ІС на транспорті</p>	<p>Programming Languages, Technologies And Paradigms</p> <p><u>1</u></p>	<p>1. Introduction</p> <p>1. Motivation.</p> <p>2. Basic concepts (types, polymorphism, reflection,...).</p> <p>3. Main programming models: imperative, functional, logical, OO, concurrent.</p> <p>4. Other paradigms. Interaction-based, emerging.</p> <p>2. Fundamentals of programming languages</p> <p>1. Syntax and static semantics of programming languages.</p> <p>2. Dynamic semantics of programming languages. Styles of semantic definition. Operational semantics.</p> <p>3. Semantic properties: Correction, Completeness, Equivalence. Specification versus scheduling.</p> <p>4. Implementation of programming languages: virtual machines and intermediate languages.</p> <p>3. Functional Paradigm</p>	<p>Курс передбачає наявність у студентів базових знань з програмування. Зокрема, практичні роботи припускають, що студент вільно володіє базовим програмуванням на Java (створення класів, атрибутів, методів, циклів тощо).</p>

	<p><i>ЗМ 2. Класифікація за допомогою вирішальних функцій. Розпізнавання в просторі ознак. Гіпотеза компактності</i></p> <p>3. Метод головних компонент. Лінійний дискримінант Фішера. Класифікація за допомогою функцій відстані. Способи стандартизації ознак і векторів-образів. Способи вимірювання відстаней між векторами ознак. Способи визначення відстані між вектором-образом і класом.</p> <p>4. Метод найближчих сусідів. Вирішальне правило, засноване на методі найближчих сусідів. Метод оцінювання за допомогою апроксимації функції густини. Розбиття класу на кластери (векторне квантування). Алгоритм К-внутрішньогрупових середніх. Постановка задачі кластеризації. Алгоритм простої розстановки центрів кластерів.</p> <p><i>ЗМ 3. Алгоритм ISODATA інтеративний аналіз даних. Нейронні мережі і проблеми розпізнавання.</i></p> <p>5. Алгоритм, заснований на методі просівання. Алгоритм максиміальної відстані. Алгоритм ІСОМАД (ISODATA). Метод (машина) опорних векторів. Лінійно роздільний випадок. Лінійно нероздільний випадок</p> <p>6. Поняття перцептрона. Алгоритм навчання перцептрона. Збіжність алгоритму перцептрона. Алгоритм</p>			<ol style="list-style-type: none"> 1. Brief introduction to functional notation. 2. Types in functional programming. 3. Polymorphism in the languages functions: genericity, coercion and overload. 4. Operational semantics: reduction and evaluation. 5. Advanced features. <p>4. Logical Paradigm</p> <ol style="list-style-type: none"> 1. Introduction to logical programming 2. Syntax of logic programs 3. The computer model of logical programming <p>5. Support Technologies and Tools</p> <ol style="list-style-type: none"> 1. Motivation 2. Support Technologies 3. Symbolic evaluation in the analysis, debugging and validation of programs. 4. Assertions for program analysis <p>Understanding the essential aspects of the different programming paradigms and the implications derived from the use and design of a language facilitates the subsequent learning of new languages, as well as their design. It also provides the necessary basis for an adequate choice and use (possibly combined) of languages and paradigms in tasks of analysis, design, construction and maintenance of applications in a robust, safe and efficient way. The objective of the course is to introduce the fundamentals, the supporting technologies and the main applications of the key paradigms in which the current programming languages are framed. The fundamentals and concepts of programming languages will be presented and illustrated using different languages.</p>	
--	--	--	--	---	--

	<p>навчання шару персептронів розділенню декількох класів. Ідеологія нейроінформатики. Елементи нейронних мереж. Архітектура нейронних мереж. Математичні можливості нейронних мереж</p> <p><i>ЗМ 4. Базові математичні задачі, вирішувані нейронними мережами. Навчання багатозарової НМ СРО методом зворотного розповсюдження помилки</i></p> <p>7. Основні алгоритми навчання нейронних мереж . Алгоритми навчання одного нейрона. Алгоритм навчання Хебба. Персептронний метод навчання. Адаптивне</p> <p>8. Алгоритм і мережа Кохонена. Мережі асоціативної пам'яті. Алгоритм і мережа Хопфілда. Алгоритм і мережа Хеммінга. Метод потенційних функцій</p>				
/Мікропроцесорні системи на транспорті					
<p>Технології розподілених систем та паралельних обчислень</p> <p>Ковальчук Оксана Петрівна</p>	<p><i>ЗМ 1. Основи паралельних обчислень.</i></p> <p>1. Класифікація систем паралельної обробки даних. Моделі зв'язку та архітектури пам'яті. Симетричні мультипроцесорні системи SMP. Паралельні векторні системи (PVP). Системи з масовим паралелізмом (MPP). Системи з неоднорідним доступом до пам'яті (NUMA). Кластерні системи</p>				

	<p>2. Системи з масовим паралелізмом (MPP). Системи з неоднорідним доступом до пам'яті (NUMA). Кластерні системи</p> <p><i>ЗМ 2. Моделювання та аналіз паралельних обчислень</i></p> <p>3. Модель обчислень у вигляді графа "операції - операнди". Опис схеми паралельного виконання алгоритму. Визначення часу виконання паралельного алгоритму. Показники ефективності паралельного алгоритму.</p> <p>4. Основні принципи OpenMP. Принципова схема програмування в OpenMP. Особливості реалізації директив OpenMP. Директиви shared, private і default. Директиви firstprivate і lastprivate. Директива if. Директива reduction. Директива coruіn. Директива for. Директива do. Директива workshare. Директива sections.</p> <p><i>ЗМ 3. Основи роботи з MPI.</i></p> <p>5. Ініціалізація паралельної частини програми. Завершення паралельної частини програми. Загальна схема MPI-програми на мові Сі. Основні функції MPI.</p> <p>6. Передача / прийом повідомлень з блокуванням. Передача повідомлення з буферизацією. Блокуючий прийом в буфер. Процедура MPI_PROBE. Послідовний обмін повідомленнями між двома процесами. Передача / прийом повідомлень без блокування.</p>				
--	---	--	--	--	--

	<p>Відкладені запити на взаємодію. Тупикові ситуації (deadlock). <i>ЗМ 4. Задачі розробки паралельних алгоритмів.</i></p> <p>7. Постановка завдання. Класифікація алгоритмів по типу паралелізму. Загальна схема етапів розробки паралельних алгоритмів. Задача філософів, що обідають. Задача постачальника-споживача. Задача читачів-письменників. Задача сплячого парикмахера.</p> <p>8. Виявлення паралелізму алгоритмів на основі аналізу графів. Постановка завдання розпаралелення. Побудова графа алгоритму обчислення перехідного процесу. Побудова й перетворення матриці 7 слідування. Виявлення логічно несумісних операторів.</p>				
/Крос-платформне програмування					

Informatics Engineering

This bachelor's degree offers two career paths at different UPV schools:



Alcoy Campus Site (Alcoy)

Higher Polytechnic School of
Alcoy



Vera Campus Site (Valencia)

School of Informatics