

Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра математичного моделювання та штучного інтелекту (№ 304)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)

Дмитро ЧУМАЧЕНКО
(ім'я та ПРИЗВИЩЕ)

« 21 » 08 2025 р.

**СИЛАБУС ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Проектування програмного забезпечення
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 122 «Комп'ютерні науки»
(код і найменування спеціальності)

Освітня програма: «Інтелектуальні системи та технології»
(найменування освітньої програми)

Рівень вищої освіти: *перший (бакалаврський)*

Силабус уведено в дію з 01.09.2025

Харків – 2025 р.

Розробник: Базілевич К.О., доц. каф. 304, к. т. н., доц.
(прізвище та ініціали, посада, науковий ступінь і вчене звання)


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри № 304
математичного моделювання та штучного інтелекту
(назва кафедри)

Протокол № 1 від « 21 » серпня 2025 р.

В. о. завідувача кафедри к.ф.-м.н., доцент
(науковий ступінь і вчене звання)


(підпис)

Олексій КАРТАШОВ
(ім'я та ПРІЗВИЩЕ)

Погоджено з представником здобувачів освіти:

Здобувач першого (бакалаврського)

рівня вищої освіти за ОП

«Інтелектуальні системи та технології»


(підпис)

Катерина ПИЛИПЕНКО
(ім'я та ПРІЗВИЩЕ)

1. Загальна інформація про викладача



ПІБ: *Базілевич Ксенія Олексіївна*

Посада: *доцент каф. 304*

Науковий ступінь: *канд. техн. наук*

Вчене звання: *доцент*

Перелік дисциплін, які викладає:

Інтелектуальний аналіз даних; Проєктування програмного забезпечення; Сучасні методи проєктування програмного забезпечення

Напрями наукових досліджень:

Інтелектуальний аналіз даних

Контактна інформація:

k.bazilevych@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	6
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	6 кредитів ЄКТС / 180 годин (64 аудиторних, з яких: лекції – 32, практичні – 48; СРЗ – 100);
Види навчальної діяльності	Лекції та практичні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – іспит
Пререквізити	«Програмування та алгоритмічні мови», «Об'єктно-орієнтоване програмування»

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – забезпечення базової профілюючої підготовки за фахом, формування теоретичних знань та практичних навичок у галузі проєктування інформаційних систем. Здобуті у процесі вивчення дисципліни знання є базою для вивчення дисциплін професійно-орієнтованого циклу.

Завдання – формування у студентів базових системних понять і навичок, цілісного бачення інформаційної системи, посилення міждисциплінарних зв'язків, розвиток системного мислення, без яких неможливе ефективне використання інформаційних технологій.

Компетентності, які набуваються:

Інтегральна компетентність:

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності (ЗК)

Після закінчення цієї програми здобувач освіти буде здатен:

- до абстрактного мислення, аналізу та синтезу (ЗК1);
- застосовувати знання у практичних ситуаціях (ЗК2);
- розуміти предметну область та бути здатним до професійної діяльності (ЗК3);
- спілкуватися державною мовою як усно, так і письмово (ЗК 4);
- вчитися й оволодівати сучасними знаннями (ЗК 6);
- здатність працювати в команді (ЗК 9).

Спеціальні компетентності (СК)

Після закінчення цієї програми здобувач освіти буде здатен:

– до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (СК 3);

– проєктувати 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 механізмами управління (СК 8).

Програмні результати навчання (ПРН):

– Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук (ПРН 1);

– Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт) (ПРН 11);

– 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 (ПРН 14).

4. Зміст навчальної дисципліни

Модуль 1.

Змістовий модуль 1. Основи проєктування програмного забезпечення інформаційних систем

Тема 1. Вступ до дисципліни

Стисла анотація: Поняття інформаційної системи (ІС). Класи ІС. Структура багатокористувальницької та однокористувальницької ІС, малої та корпоративної ІС, локальної і розподіленої ІС, склад і призначення підсистем. Основні особливості сучасних проєктів ІС. Програмне забезпечення та відмінність від інформаційних систем.

Теми лекцій і практичних занять: «Попередній аналіз предметної області».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Тема 2. Аналіз вимог

Стисла анотація: Аналіз вимог. Типи вимог: функціональні вимоги та нефункціональні вимоги. Похідні вимоги. Рівні вимог. Системні та програмні вимоги. Функціональні, нефункціональні вимоги та характеристики продукції. Особливості продукту. Списки вимог. Прототипи (досвідчені зразки). Сценарії використання. Специфікація вимог програмного забезпечення. Бачення проєкту (vision). Технічне завдання.

Теми лекцій: «Аналіз вимог»

Теми практичних занять: «Розробка документу Бачення»

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Тема 3. Аналіз і моделювання функціональної області впровадження ІС

Стисла анотація: Основні поняття організаційного бізнес-моделювання. Дерево цілей і стратегії їх досягнення. Статичний і динамічний опис предметної області. Повна бізнес-модель. Шаблони організаційного бізнес-моделювання. Побудова організаційно-функціональної структури. Інформаційні технології організаційного моделювання. Результати передпроектного обстеження. Методології моделювання предметної області. Структурна модель предметної області. Об'єктна структура. Функціональна структура. Структура управління. Організаційна структура. Функціонально-орієнтовані і об'єктно-орієнтовані методології опису предметної області. Функціональна методика IDEF. Case-засоби для моделювання ділових процесів. Інструментальні середовища. Принципи побудови моделі IDEF0: контекстна діаграма, суб'єкт моделювання, мета і точка зору. Діаграми IDEF0: контекстна діаграма, діаграми декомпозиції, діаграми дерева вузлів, діаграми тільки для експозиції (FEO). Діаграми потоків даних (Data Flow Diagramming): роботи, зовнішня суть (посилання), потоки робіт, сховища даних. Метод опису процесів IDEF3: роботи, зв'язки, об'єкти посилань, перехрестя.

Теми лекцій і практичних занять: «Моделювання бізнес-процесів у системі»

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Тема 4. Життєвий цикл програмного забезпечення ІС

Стисла анотація: Поняття життєвого циклу ПЗ ІС. Процеси життєвого циклу: основні, допоміжні, організаційні. Зміст і взаємозв'язок процесів життєвого циклу ПЗ ІС. Моделі життєвого циклу: каскадна, модель з проміжним контролем, спіральна. Регламентація процесів проєктування у вітчизняних і міжнародних стандартах.. Стадії і етапи процесу канонічного проєктування ІС. Цілі і завдання передпроектної стадії створення ІС. Моделі діяльності організації. Склад робіт на стадії технічного і робочого проєктування. Склад проектної документації.

Теми лекцій: «Життєвий цикл програмного забезпечення ІС»

Теми практичних занять: «Розробка документу Технічне завдання»

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Тема 5. Сучасні методології розробки програмного забезпечення

Стисла анотація: Загальне уявлення про процес. Початкова фаза проекту. Уточнення. Ризики, пов'язані з вимогами. Технологічні ризики.

Базова архітектура. Конструювання. Введення в дію. Робочі процеси. Знайомство з Agile-технологіями.

Теми лекцій: «Ознайомлення із сучасними методологіями управління проектами у сфері розробки програмного забезпечення».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Модульний контроль 1

Змістовий модуль 2. Проєктування ІС із застосуванням UML

Тема 6. Етапи проєктування ІС із застосуванням UML

Стисла анотація: Діаграми в UML. Класи і стереотипи класів. Асоціативні класи. Основні елементи діаграм взаємодії – об'єкти, повідомлення. Діаграми станів: початкового стану, кінцевого стану, переходи. Вкладеність станів. Моделювання бізнес-прецедентів, розробка моделі бізнес-об'єктів, розробка концептуальної моделі даних, розробка вимог до системи, аналіз вимог і попереднє проєктування системи, розробка моделей даних і додатків, проєктування фізичної реалізації системи.

Теми лекцій і практичних занять: «Розробка діаграм варіантів використання», «Розробка діаграм взаємодії», «Розробка діаграми класів»

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Модульний контроль 2

Змістовий модуль 3. Тестування програмного забезпечення

Тема 7. Основні поняття тестування

Стисла анотація: Якість. Значення якості для підприємства. Процес (стадії) розробки ПЗ за схемою. Роль QA на всіх етапах проекту. Quality assurance vs. testing – відмінність понять. Поняття тестування ПЗ. Його принципи. Основні терміни тестування. Підходи до обґрунтування істинності формул і програм і їх зв'язок з тестуванням. Основні проблеми тестування, завдання вибору кінцевого набору тестів.

Теми лекцій і практичних занять: «Тестування».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання індивідуальних завдань відповідно варіанту, оформлення звітів з практичних робіт.

Модульний контроль 3

5. Індивідуальні завдання

Виконання розрахункової роботи на тему «Прототипи програмного забезпечення та їх використання».

Необхідно розробити робочий прототип інформаційної системи за заданим варіантом (або за власною темою, узгодженою з викладачем).

Виконання курсової роботи за індивідуальним варіантом «Проектування програмного забезпечення».

Провести проектування інформаційної системи за заданим варіантом (або за власною темою, узгодженою з викладачем) з урахуванням тем, вивчених на курсі. Курсова робота має містити:

1. Проведений попередній аналіз предметної області.
2. Розроблене технічне завдання.
3. Розроблену модель бізнес-процесів у системі.
4. Розроблені діаграми: варіантів використання, взаємодії, класів
5. Звіт з проведеного тестування системи.
6. Програму (веб-сайт), яку було розроблено.

6. Методи навчання

1. Пояснювально-ілюстративний (інформаційно-рецептивний) метод: та метод проблемного виконання (лекція).

2. Репродуктивний (лабораторні роботи).

3. Частково-пошуковий (евристичний) та дослідницький: (самостійна робота).

4. Дисципліна «Проектування програмного забезпечення» передбачає лекційні (в т. ч. з використанням мультимедійного обладнання) і практичні заняття під керівництвом викладача та самостійну роботу студента за підручниками і матеріалами, що забезпечує закріплення теоретичних знань, сприяє набуттю практичних навичок і розвитку самостійного наукового мислення. Передбачено регулярні індивідуальні консультації.

7. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних (виконання практичних робіт) і підсумкових контролів (захист змістовного модуля, іспит).

Поточний контроль має на меті систематично відстежувати, наскільки студент розуміє та засвоює матеріал, вміє виконувати практичні роботи, самостійно вирішувати завдання, готувати звіти та представляти результати роботи в усній або письмовій формі.

Підсумковий контроль покликаний оцінити глибину знань студента з усього курсу, його розуміння зв'язків між різними темами, здатність застосовувати отримані знання на практиці та самостійно вирішувати задачі, які виникають в рамках дисципліни.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Виконання і захист практичних робіт	0...5	4	0...20
Модульний контроль	0...20	2	0...40
Змістовний модуль 2			
Виконання і захист практичних	0...5	3	0...15
Модульний контроль	0...5	1	0...5
Змістовний модуль 3			
Виконання і захист практичних	0...5	1	0...5
Модульний контроль	0...5	1	0...5
Виконання і захист РР	0...10	1	0...10
Усього за семестр			0...100

Семестровий контроль (*іспит*) проводиться у разі відмови здобувача освіти від балів підсумкового контролю й за наявності допуску до *іспиту*. Під час складання семестрового *іспиту* здобувач освіти має можливість отримати максимум 100 балів. Білет для *іспиту* складається з двох теоретичних і одного практичного питань. У першому та другому запитанні студент має продемонструвати теоретичні знання. У третьому запитанні – показати навички проєктування інформаційної системи. Максимальна кількість балів за кожне запитання та їх суми показані в табл. 8.2.

Таблиця 8.2 – Шкали оцінювання відповідей на запитання білету

Складові білету	Складові оцінки	Бали за одне питання	Сумарна кількість балів
Пункт 1. Теоретичне запитання	Надано визначення поняття	0...10	0...20
	Наведено власний приклад використання	0...10	
Пункт 2. Теоретичне запитання	Надано визначення поняття	0...10	0...20
	Наведено власний приклад використання	0...10	
Пункт 3. Практичне запитання	Виконано проєктування фрагменту інформаційної системи за заданими умовами	0...60	0...60
Підсумкова оцінка за іспит			0...100

Таблиця 8.3 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача освіти протягом семестру

Задовільно (60-74): мати володіння базовими знаннями та навичками, необхідними для досягнення програмних результатів навчання з проєктування програмного забезпечення; виконати практичні роботи, що охоплюють основні елементи курсу; написати модульні роботи, що демонструють розуміння ключових концепцій та технік проєктування програмного забезпечення; мати розуміння основних етапів проєктування програмного забезпечення, знати застосування різних методологій проєктування та вміти оцінювати їхню ефективність для конкретного практичного прикладу.

Добре (75-89): мати вільне володіння основними концепціями проєктування програмного забезпечення та вміти застосовувати їх на практиці, успішно виконувати більшість практичних завдань з аналізу вимог, моделювання бізнес-процесів (IDEF0, DFD, IDEF3) та проєктування архітектури ІС, демонструвати розуміння різних моделей життєвого циклу ПЗ та сучасних методологій розробки, бути здатним аналізувати та обґрунтовувати вибір певних підходів до проєктування..

Відмінно (90-100): виявляти глибоке і всебічне розуміння всіх концепцій та принципів проєктування програмного забезпечення, вільно володіти різноманітними інструментами та методами моделювання (включаючи UML), демонструвати здатність розробляти комплексні проєктні рішення, враховуючи функціональні та нефункціональні вимоги, вміти критично аналізувати різні підходи до розробки, проявляти ініціативу, творчий підхід при виконанні складних завдань, самостійно вирішувати нестандартні проблеми, бути здатним оцінювати та покращувати якість розробленого програмного забезпечення.

9. Політика навчального курсу

Відвідування занять. З огляду на інтерактивний формат курсу, відвідування практичних занять є обов'язковим. Студенти, які не можуть регулярно відвідувати практичні, повинні протягом тижня узгодити з викладачем індивідуальний графік відпрацювання пропущених занять. Пропущені заняття відпрацьовуються на найближчій консультації протягом тижня після пропуску у формі співбесіди за питаннями з плану заняття. В

окремих випадках дозволяється відпрацювання у письмовій формі шляхом виконання індивідуального письмового завдання.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>). Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu/ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/>).

10. Методичне забезпечення

Увесь науково методичний комплект з дисципліни розміщено на освітньому порталі Mentor: <https://mentor.khai.edu/course/view.php?id=5043>

11. Рекомендована література

Базова

1. С. Канер. Тестування програмного забезпечення. Фундаментальні концепції менеджменту бізнес-додатків: Пер. с англ. – К. : Видавництво «ДіаСофт», 2001. – 544 с.
2. С. В. Баран. Розробка програмного забезпечення з використанням патернів проектування: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. – 203 с.
3. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, Third Edition. Addison Wesley, 2012.
4. Міністерство освіти і науки України. Методичні рекомендації щодо розроблення стандартів вищої освіти. – Наказ МОН України від 01.06.2024 р. № 272. Режим доступу: <https://zakon.rada.gov.ua/go/v0022729-24>
5. Національне агентство із забезпечення якості вищої освіти. Роз'яснення щодо застосування Критеріїв оцінювання якості освітньої програми. – Київ, 2024. Режим доступу: <https://naqa.gov.ua>
6. Міністерство освіти і науки України. Затверджені стандарти вищої освіти: офіційний перелік. – Київ: МОН, 2024. Режим доступу: <https://mon.gov.ua/osvita-2/vishcha-osvita-ta-osvita-doroslikh/naukovo-metodichna-rada-ministerstva-osviti-i-nauki-ukraini/zatverdzeni-standarti-vishchoi-osviti>

Допоміжна

1. Grand M. Patterns in Java. A Catalog of Reusable Design Patterns Illustrated with UML – Wiley & Sons, 1998. Vol. 1, - 468 p. Vol. 2, – 356 p.
2. Starr L. Executable UML. How to build class models - Prentice Hall PTR, 2002. – 418 p.
3. Wampler B.E. The Essence Object-Oriented Programming with Java and UML - Addison-Wesley, 2002, – 290 p.
4. Powell T., Brown C. Learning Web Design. 5th Edition. O'Reilly Media. 2018. URL: <https://www.oreilly.com/library/view/learning-web-design/9781491960196/>.

12. Інформаційні ресурси

1. W3C. HTML5: A vocabulary and associated APIs. Recommendation. 28 October 2014. URL: <https://www.w3.org/TR/html5/>. WHATWG. HTML Living Standard. URL: <https://html.spec.whatwg.org/>.
2. Mozilla Developer Network. HTML Reference. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>. W3Schools. HTML Tutorial. URL: <https://www.w3schools.com/html/>.

3. Duckett J. HTML & CSS: Design and Build Websites – Online Samples. 2011. URL: <https://htmldog.com/guides/html/>.
4. HTML.com. The Modern HTML5 Tutorial. URL: <https://html.com/>.
5. Lawson B., Sharp R. Introducing HTML5. A Book Apart. 2010. URL: <https://abookapart.com/products/introducing-html5>.
6. freeCodeCamp. Responsive Web Design Certification. URL: <https://www.freecodecamp.org/learn/responsive-web-design/>.
7. MDN Web Docs. HTML Global Attributes. URL: https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes.
8. W3C. Cascading Style Sheets Level 2 Revision 1 (CSS2.1) Specification. 2011. URL: <https://www.w3.org/TR/CSS2/>.
9. W3C. CSS3 Selectors. Recommendation. 2001. URL: <https://www.w3.org/TR/selectors-3/>.
10. Mozilla Developer Network. CSS Reference. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
11. W3Schools. CSS Tutorial. URL: <https://www.w3schools.com/css/>.
12. CSS-Tricks. A Complete Guide to Flexbox. 2018. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>.
13. CSS-Tricks. A Complete Guide to Grid. 2018. URL: <https://css-tricks.com/snippets/css/complete-guide-grid/>.
14. Meyer E. CSS Reset. 2007. URL: <https://meyerweb.com/eric/tools/css/reset/>.
15. Andrew R. The New CSS Layout. 2018. URL: <https://rachelandrew.co.uk/the-new-css-layout/>.
16. MDN Web Docs. CSS Box Model. URL: https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model.
17. ECMA-262. ECMAScript 2021 Language Specification. ECMA International. 2021. URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>.
18. Godot docs. URL: <https://docs.godotengine.org/uk/4.x/index.html>.
19. Godot community. URL: <https://forum.godotengine.org/>.
20. Godot forums. URL: <https://godotforums.org/>.
21. Екстремальне програмування. URL: <https://worksection.com/ua/blog/extreme-programming.html>.
22. Agile модель. URL: <https://simplentrec.com/agile-model-sdlc/>.
23. Міністерство освіти і науки України: Інформаційна безпека бізнесу. URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/.
24. The Power of Figma as a Design Tool. URL: <https://www.toptal.com/designers/ui/figma-design-tool>.
25. Figma Features. URL: <https://www.figma.com/legal/core-features>.
26. Figma Learn. URL: <https://help.figma.com/hc/en-us/articles/30848209492887-Course-overview-Figma-Design-for-beginners-2025>.
27. FL Studio Learn. URL: <https://www.image-line.com/fl-studio-learning/>.

28. FL Studio Plugins & Instruments. URL: <https://www.image-line.com/fl-studio/plugins/>.
29. FL Studio. URL: https://en.wikipedia.org/wiki/FL_Studio.