

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана  
з навчальної роботи

 Кашпур О.Ф.

«30» серпня 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
ІНСТРУМЕНТАЛЬНІ СЕРЕДОВИЩА  
ТА ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ**

для студентів

галузь знань **12 «Інформаційні технології»**  
(шифр і назва)

спеціальність **122 «Комп'ютерні науки»**  
(шифр і назва спеціальності)

освітній рівень **бакалавр**  
(молодший бакалавр, бакалавр, магістр)

освітня програма **«Інформатика»**  
(назва освітньої програми)

вид дисципліни **обов'язкова**

Форма навчання

**заочна**

Навчальний рік

**2019/2020**

Семестр

**6**

Кількість кредитів ECTS

**5**

Мова викладання, навчання

та оцінювання

**українська**

Форма заключного контролю

**іспит**

Викладачі: к.ф.-м.н., доц. Омельчук Л.Л. (лекції, лабораторні заняття),

к.п.н., асистент Русіна Н.Г. (лабораторні заняття)

Пролонговано: на 20 20/2021 н.р.   «30» серпня 2020 р.

на 20    /20    н.р. (    ) «    »    20    р.

(підпис, ПІБ, дата)

Розробник: Омельчук Людмила Леонідівна, к.ф.-м.н., доцент кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»

(підпис)

Нікітченко М.С.

(прізвище та ініціали)

Протокол № 1 від « 28 » 08 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від « 30 » 08 2019 року № 1

Голова науково-методичної комісії

(підпис)

Омельчук Л.Л.

(прізвище та ініціали)

**1. Мета дисципліни** – засвоєння знань з інструментальних середовищ та технологій програмування. Оволодіння базовими навичками проектування програмних систем, набуття навичок використання інструментальних середовищ програмування, та використання технологій роботи з даними та технологій створення веб-додатків.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни:**

1. *Знати:* основні поняття об'єктно-орієнтованого програмування, основні етапи життєвого циклу ПС, шаблони, антишаблони та принципи об'єктно-орієнтованого проектування програмного забезпечення.

2. *Вміти:* застосовувати на практиці інструментальні програмні засоби проектування та розробки програмного забезпечення.

3. *Володіти елементарними навичками:* програмування мовою C#.

**3. Анотація навчальної дисципліни:**

Навчальна дисципліна “Інструментальні середовища та технології програмування” є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти *галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, освітньо-професійної програми – „Інформатика”*.

Дана дисципліна є обов’язковою навчальною дисципліною за *програмою “Інформатика”*.

Викладається у 4 семестрі 2 курсу в **обсязі – 150 год.**

*(5 кредитів ECTS)* зокрема: *лекції – 7 год., лабораторні – 7 год., консультації – 1 год., самостійна робота – 135 год.* У курсі передбачено **2 частини** та **2 контрольні роботи**. Завершується дисципліна – **іспитом в 4 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен

**знати** основи реляційних баз даних та мови запитів SQL, технології розробки інформаційних програмних систем, принципи роботи технологій доступу до даних на прикладі ADO.Net, основи HTML, CSS, JavaScript, базові елементи програмної інженерії, принципи роботи технологій створення веб-застосунків на прикладі ASP.Net.

**вміти** працювати з технологією ADO.Net на автономному рівні, працювати з технологією ADO.Net Entity Framework, працювати з технологією ASP.Net.

Для допуску до дисципліни „Інструментальні середовища та технології програмування” освітньо-професійної програми «Інформатика» студент повинен опанувати компетентності та результати навчання, які надає дисципліна „Об’єктно-орієнтоване програмування” освітньо-професійної програми «Інформатика». Дисципліна „Інструментальні середовища та технології програмування” пов’язана з дисципліною „Бази даних та інформаційні системи”. Дисципліна „Інструментальні середовища та технології програмування” є базовою для засвоєння дисципліни «Системного програмування».

#### 4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетенцій) на рівні новітніх досягнень у програмуванні, відповідно до освітньої кваліфікації «Бакалавр з комп'ютерних наук». Зокрема, розвивати:

- здатність застосовувати знання у практичних ситуаціях;
- здатність оцінювати та забезпечувати якість виконуваних робіт;
- 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 механізмами управління;
- здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, сховища даних і бази знань, для забезпечення обчислювальних потреб багатьох користувачів, обробки транзакцій, у тому числі на хмарних сервісах;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність використовувати інтелектуальні інформаційні технології;
- здатність реалізовувати фази та ітерації життєвого циклу створення програмних систем на основі моделей та методів розробки програмного забезпечення.

#### 5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	Знати основи реляційних баз даних та мови запитів SQL, технології розробки інформаційних програмних систем	Лекція, лабораторна робота, робота	Контрольна робота, іспит	17%
PH1.2	Знати принципи роботи технологій доступу до даних на прикладі ADO.Net	Лекція, лабораторна робота, робота		
PH1.3	Знати основи HTML, CSS, JavaScript	Лекція, лабораторна робота, робота		17% 15%
PH1.4	Знати базові елементи програмної інженерії	Лекція, лабораторна робота, робота		
PH1.5	Знати принципи роботи технологій створення веб-застосунків на прикладі ASP.Net	Лекція, лабораторна робота, робота		
PH2.1	Вміти працювати з технологією ADO.Net на автономному рівні	лабораторна робота, самостійна робота	Захист лабораторної роботи, іспит	33%
PH2.2	Вміти працювати з технологією ADO.Net Entity Framework	лабораторна робота, самостійна робота	Захист лабораторної роботи, іспит	33%
PH2.3	Вміти працювати з технологією ASP.Net	лабораторна робота, самостійна робота	Захист лабораторної роботи	33%

**6. Співвідношення результатів навчання дисципліни із програмними результатами навчання**

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 1.4	РН 1.5	РН 2.1	РН 2.2	РН 2.3
<b>Програмні результати навчання</b>								
<i>(з опису освітньої програми)</i>								
ПРН9. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них.	+	+				+		+
ПРН11. Володіти навичками використання методології управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти готувати проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, креативний бриф, угоду, договір, контракт та ін.).		+	+	+	+		+	

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### - семестрове оцінювання:

1. Контрольна робота (тест): РН 1.1., РН 1.2 — 10 балів/6 балів.
2. Контрольна робота (тест): РН1.3, РН 1.4., РН 1.5 - 11 балів/6.6 балів.
3. Лабораторна робота 1 (проект): РН 2.1 – 13 балів/7.8 балів.
4. Лабораторна робота 2 (проект): РН 2.2 – 13 балів/7.8 балів.
5. Лабораторна робота 3 (проект): РН 2.3 – 13 балів/7.8 балів.

#### - підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН1.4, РН1.5, РН2.1, РН2.2;
- форма проведення і види завдань: письмова робота.

#### Види завдань:

##### Структура екзаменаційної роботи та критерії оцінювання:

1. Теоретичне запитання з БД (РН1.1).
2. Теоретичне запитання з SQL (РН1.1).
3. Теоретичне запитання з технологій доступу до даних (РН1.2).
4. Теоретичне запитання ADO.NET (РН1.2).
5. Теоретичне запитання ADO.NET(РН1.2).
6. Теоретичне запитання з Entity Framework, або програмної інженерії (в залежності від варіанту) (РН1.2, РН1.4).
7. Тестове запитання з HTML, CSS, JavaScript (РН1.3).
8. Письмове завдання з ASP.NET (РН1.5).
9. Тестове запитання з Entity Framework (РН1.2).
10. Тестове запитання з HTML, CSS, JavaScript (РН1.3).
11. Тестове запитання з LINQ (РН1.4).
12. Письмове запитання з ASP.NET (РН1.5).
13. Письмове з БД (розробити діаграму БД) (РН1.1).

#### Критерії оцінювання екзаменаційної роботи

Завдання	Вид завдання	Максимальний відсоток (бал)	Всього відсотків (балів)
Завдання 1, 2, 3, 4, 5, 6	Письмове запитання з курсу	по 6% (1.6 балів)	36% (14.4 бали)
Завдання 7, 9, 10	Тестові завдання	по 5% (2 бали)	15% (6 балів)
Завдання 8	Тестове завдання	9% (3.6 балів)	9% (3.6 балів)
Завдання 11	Тестове завдання	10% (4 бали)	10% (4 бали)
Завдання 12, 13	Письмове завдання	по 15% (6 балів)	30% (12 балів)
<b>Всього</b>			<b>100% (40 балів)</b>

#### Запитання для підготовки до іспиту

1. База даних. Система керування БД.
2. Моделі даних.
3. Поняття база даних (БД). Система керування базами даних (СКБД). Моделі даних. Поняття реляційної бази даних. Реляційна алгебра. Первинний ключ, зовнішні ключі.
4. Реляційна алгебра.
5. Стандарти SQL та рівні відповідності. Основи мови запитів SQL. Обробка даних (SELECT, INSERT, DELETE, UPDATE). Визначення даних (CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE VIEW).
6. Засоби для роботи з реляційними базами даних. Огляд ORM-технологій.
7. Реляційні бази даних.
8. Поняття таблиці, первинного ключа, зовнішнього ключа, кортеж, атрибут, степінь.
9. Типи даних мови SQL.
10. Вирази в SQL.
11. Запит SQL на створення таблиці. Запит SQL на зміну структури існуючої таблиці.
12. Запит SQL на видалення таблиці. Запит SQL на додавання нових рядків.
13. Запит SQL на пошук. Запит SQL на видалення рядків.
14. Запит SQL на оновлення існуючих даних.
15. Транзакція. Тригер. Представлення.
16. Технології доступу до даних. Призначення. Приклади.
17. Коротка характеристика ADO.
18. Коротка характеристика ODBC.
19. Коротка характеристика OLE DB
20. Коротка характеристика BDE
21. Коротка характеристика DAO, JDBC.
22. Рівні застосування бібліотеки ADO.NET (підключений (connected layer) автономний рівень (disconnected layer) та за допомогою технології Entity Framework).
23. Архітектура ADO.NET. Постачальники даних. Connection, Command, DataReader, DataAdapter, Parameter, Transaction. Підключення до бази даних. Команди. Виконання команди. Параметризовані об'єкти команд. Транзакції.
24. Архітектура ADO.NET.
25. Рівні застосування бібліотеки ADO.NET.
26. ORM (Object-relational Mapping).
27. Архітектура ADO.NET. DataSet.
28. Архітектура ADO.NET. DataAdapter.
29. Архітектура доступу до даних Entity Framework.
30. Модель EDM (Entity Data Model). edmx.
31. ADO.NET Entity Framework (EF). EntityClient. DbContext.
32. ADO.NET. Рівні застосування.
33. Рівні застосування бібліотеки ADO.NET (підключений (connected layer) автономний рівень (disconnected layer) та за допомогою технології Entity Framework). Автономний рівень.
34. ADO.NET. Постачальники даних.
35. ADO.NET. Команди. Виконання команд.
36. ADO.NET. Підключення до даних.
37. ADO.NET. Транзакції
38. ADO.NET. Архітектура.
39. Архітектура доступу до даних Entity Framework.
40. Підключення. Entity Data Model (EDM). Концептуальна та сутнісна моделі даних. Доступ до даних. Архітектура доступу до даних. Доступ до даних сутностей та їх зміна. Запити до концептуальної моделі даних. EntityClient, DbContext.
41. LINQ to Entities.
42. Селектори jQuery.
43. Шаблон проектування Model-view-controller (MVC, «Модель-представлення-контролер»).
44. Архітектура ASP.NET MVC 4.
45. ASP.NET MVC. Контролери.
46. ASP.NET MVC. Моделі.
47. ASP.NET MVC. Представлення.
48. Опишіть підхід до роботи з даними в Entity Framework Database First.

49. Опишіть підхід до роботи з даними в Entity Framework Model First.
50. Опишіть підхід до роботи з даними в Entity Framework Code First
51. Структура SWEBOOK.
52. Наведіть визначення програмної інженерії та інформатики. Опишіть їх зв'язок та відмінності.
53. Короткий огляд HTTP. HTML. CSS. Підключення CSS до HTML. Правила CSS. Селектори CSS. Класи. Псевдоелементи. Блоки. Списки.
54. Шаблон проектування Model-view-controller (MVC, «Модель-представлення-контролер»).
55. ASP.NET MVC. Контролери. Моделі.
56. ASP.NET MVC. Представлення. Двигун представлення Razor.
57. Програмна інженерія. Життєвий цикл програмного продукту та моделі процесу.
58. Agile методи. Принципи та основні ідеї.
59. Життєвий цикл програмного продукту та моделі процесу; моделі оцінки процесу; метрики процесу розробки програмного забезпечення.
60. Еволюція програм: супровід програм, характеристики супроводжуваного програмного забезпечення (ПЗ), реінженірінг, успадковані системи, повторне використання ПЗ
61. Масштабованість, продуктивність і ефективність інформаційних технологій та систем. Перспективи розвитку.
62. Розробка ПЗ: об'єктно-орієнтований аналіз і проектування; проектування для повторного використання; шаблони проектування; програмні оточення; програмні засоби тестування.

**Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів. Студент допускається до іспиту за умови виконання 70% передбачених планом лабораторних робіт.**

## **7.2 Організація оцінювання:**

### **Терміни проведення форм оцінювання:**

1. Контрольна робота (тест): до другого лекційного заняття.
2. Контрольна робота (тест): до другого лекційного заняття.
3. Лабораторна робота 1 (проект): до другого лекційного заняття.
4. Лабораторна робота 2 (проект): до другого лекційного заняття.
5. Лабораторна робота 3 (проект): до другого лекційного заняття.

Студент має право на одне перескладання контрольної роботи із можливістю отримання максимально 5 балів за кожен раз. Термін перескладання визначається викладачем.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

## **7.3 Шкала відповідності оцінок**

<b>Відмінно / Excellent</b>	90-100
<b>Добре / Good</b>	75-89
<b>Задовільно / Satisfactory</b>	60-74
<b>Незадовільно / Fail</b>	0-59

## 8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаборат. заняття	Сам. р-та
<b>Частина 1. Огляд та застосування ORM (Object-relational mapping, Об'єктно-реляційне відображення)-технологій</b>				
1.	<b>Тема 1. Поняття реляційної бази даних. Огляд сучасних СУБД.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 1.		1	5
2.	<b>Тема 2. Основи мови запитів SQL. Обробка даних (SELECT, INSERT, DELETE, UPDATE). Визначення даних (CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE VIEW). Керування транзакціями (COMMIT, ROLLBACK).</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 1.			5
3.	<b>Тема 3. Засоби для роботи з реляційними базами даних. Огляд ORM-технологій.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 1.		1	10
4.	<b>Тема 4. Бібліотека ADO.NET. Архітектура ADO.NET. Рівні застосування бібліотеки ADO.NET. Підключений (connected layer) рівень.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 1.	1		10
5.	<b>Тема 5. Технологія ADO.NET. Автономний рівень (disconnected layer). Застосування LINQ to DataSet та SQL.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 1.	1		10
6.	<b>Тема 6. Архітектура доступу до даних Entity Framework. Застосування LINQ to Entities.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 2.	1		10
7.	<b>Тема 7. Короткий огляд HTTP. Основи HTML, CSS.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 2.	1		5
8.	<b>Тема 8. Основи JavaScript.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 2.		1	5
9.	<b>Тема 9. Короткий огляд jQuery. Підтримка AJAX.</b> Самостійна робота: Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 2.		1	5
	Контрольна робота 1			5
Контроль за підсумками лабораторної роботи 1				
Всього по частині 1		4	4	70
<b>Частина 2. Розробка web-застосувань. Елементи інженерії програмних систем</b>				
10.	<b>Тема 10. Шаблон проектування Model-view-controller (MVC, «Модель-представлення-контролер»). Та його застосування в ASP.NET MVC.</b> Самостійна робота: Опрацювання лекційного матеріалу та	1		5

	власного проекту лабораторного практикуму № 2.			
11.	<b>Тема 11. ASP.NET MVC. Контролери. Моделі.</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.		1	5
12.	<b>Тема 12. ASP.NET MVC. Представлення. Двигун представлення Razor.</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.	1		5
13.	<b>Тема 13. ASP.NET Core MVC. Контролери. Моделі.</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.			10
14.	<b>Тема 14. ASP.NET Core MVC. Представлення.</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.	1		5
15.	<b>Тема 15. Елементи інженерії програмних систем (ПС). Життєвий цикл програмного продукту та моделі процесу.</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.		1	10
16.	<b>Тема 16. Еволюція програм: супровід програм, характеристики супроводжуваного програмного забезпечення (ПЗ).</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.		1	10
17.	<b>Тема 17. Організація робіт з контролю якості ПЗ. Методи тестування. Програмні засоби тестування. Поняття про верифікацію та валідацію. Стиль та рефакторинг.</b> <i>Самостійна робота:</i> Опрацювання лекційного матеріалу та власного проекту лабораторного практикуму № 3.			10
	<i>Контрольна робота 2</i>			5
Контроль за підсумками лабораторних робіт 2 та 3				
	Всього по частині 2	3	3	65
	<b>ВСЬОГО</b>	<b>7</b>	<b>7</b>	<b>135</b>

**Загальний обсяг 150 год.**, в тому числі:

Лекцій – 7 год.

Лабораторні заняття - 7 год.

Консультації – 1 год.

Самостійна робота - 135 год.

## 9. Рекомендовані джерела:

### **Основна**

1. <http://cyb.univ.kiev.ua/uk/departments.ttp.omelchuk.html>
2. В.В. Зубенко, Л.Л. Омельчук. Програмування : навчальний посібник (гриф МОН України) / - К. : ВПЦ "Київський університет", 2011. - 623 с.
3. <http://msdn.microsoft.com/ru-RU/>
5. Лаврищева Е.М. Современные методы программирования: возможности и инструменты // Проблемы програмування. – 2006.– № 2-3. – С.60-74.
6. Буч Г. Объектно-ориентированное программирование. – Киев, Диалектика, 1993
7. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. ISBN 5-469-00599-2

8. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2

**Додаткова:**

9. Леффингуелл Д, Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный поход. . – М.: Изд. дом “Вильямс”, 20024. Д. Дайтеман и др. Программирование в среде Delphi – – Київ, 1995.

10. Ставровский А.Б, Карнаух Т.О. Перші кроки програмування.-К.:Диалектика.-2005, с.389.

11. Кнут Д. Искусство программирования.Т.1,2,3. Изд 3-е, испр. -М.СБП.К.:Вильямс, 2001.

12. Вирт Н. Систематическое программирование. Введение.- М.:Мир,1987.с.184.

13. Вирт Н. Алгоритмы и структуры данных. - М.:Мир,1989.с.263.

14. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL: полный справочник = SQL: The Complete Reference. — 3-е изд. — М.: Вильямс, 0. — 960 с. — ISBN 978-5-8459-1654-9

15. Алекс Кригель, Борис Трухнов. SQL. Библия пользователя. Язык запросов SQL = SQL Bible. — 2-е изд. — М.: Диалектика, 2009. — 752 с. — ISBN 978-5-8459-1546-7

16. К. Дж. Дейт. Введение в системы баз данных / Пер. с англ. — 8-е изд. — М.: Вильямс, 2005. — 1328 с.

**10. Додаткові ресурси:**

<https://dl.knu.ua/course/view.php?id=8379>

<https://drive.google.com/drive/u/1/folders/10HmCrg369oqF-hr225jcvxV0BX787W8z>