

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

(повне найменування вищого навчального закладу)

Кафедра програмних засобів

“ЗАТВЕРДЖУЮ”

Ректор (перший проректор)

“_____” _____ 20__ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

“ Кросплатформне програмування ”

(код і назва навчальної дисципліни)

спеціальність 121 Інженерія програмного забезпечення,
122 Комп’ютерні науки

(код і назва спеціальності)

спеціалізація _____

(назва спеціалізації)

інститут, факультет Інститут інформатики та радіоелектроніки,
Факультет комп’ютерних наук і технологій

(назва інституту, факультету)

Запоріжжя – 2016 рік

Робоча програма “Кросплатформне програмування”

(назва навчальної дисципліни)

для студентів спеціальностей 121 Інженерія програмного забезпечення, 122 Комп’ютерні науки.

“ 16 ” серпня, 2016 року – 10 с.

Розробники: (вказати авторів, їхні посади, наукові ступені та вчені звання)

Олійник Андрій Олександрович, к.т.н., доцент

Робоча програма затверджена на засіданні кафедри
програмних засобів

Протокол від. “ ” 20__ року №__

Завідувач кафедри програмних засобів

(Субботін С.О.)
(підпис) (прізвище та ініціали)

“ ” 20__ року

Схвалено методичною комісією вищого навчального закладу за спеціальністю 121 Інженерія програмного забезпечення, 122 Комп’ютерні науки та інформаційні технології

(код, назва)

Протокол від. “ ” 20__ року №__

“ ” 20__ року Голова (Касьян М.М.)
(підпис) (прізвище та ініціали)

©_____, 20__ рік

©_____, 20__ рік

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 4	Галузь знань <u>12 «Інформаційні технології»</u> (шифр і назва)	Нормативна	
Модулів – 2	Спеціальність (професійне спрямування): <u>121 Інженерія програмного забезпечення, 122 Комп'ютерні науки</u>	Рік підготовки:	
Змістових модулів – 2		4-й	–
Індивідуальне науково-дослідне завдання –		Семестр	
Загальна кількість годин – 120		7-й	–
Тижневих годин для денної форми навчання: аудиторних – 3 самостійної роботи студента – 6	Освітній-ступінь: бакалавр	Лекції	
		12 год.	4 год.
		Практичні, семінарські	
		–	–
		Лабораторні	
		26 год.	6 год.
		Самостійна робота	
		82 год.	110 год.
		Індивідуальні завдання: –	
		Вид контролю: Екзамен	

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 38/82

2. Мета та завдання навчальної дисципліни

Мета: вивчення та практичне засвоєння методів та засобів методів та засобів кроссплатформного програмування для їх застосування на процесах розробки програмних систем.

Завдання:

- сформувати знання та отримати практичні навички про кроссплатформне програмування;
- отримати уяву про характер і роль кроссплатформного програмування у сучасному світі;
- набути навичок та отримати досвід по створенню програмних додатків, що є кроссплатформними.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- особливості кроссплатформних систем;
- засоби розробки кроссплатформного програмного забезпечення;
- вимоги до систем кроссплатформної направленості;
- переваги та недоліки засобів розробки, що дозволяють створювати програмні додатки для будь-яких операційних систем;
- місце кроссплатформної розробки у веб;

вміти:

- обирати засоби розробки;
- програмувати універсальні системи;
- створювати кроссплатформні програмні продукти;
- розгортати кроссплатформні системи;
- розроблювати кроссплатформні системи під веб.

3. Програма навчальної дисципліни

Змістовий модуль 1. Вступ до кросплатформного програмування

Тема 1. Поняття про кросплатформність. Основні переваги та недоліки кросплатформного програмного забезпечення.

Тема 2. Крос-платформність на рівні апаратної платформи. Крос-платформність на рівні ОС.

Тема 3. Крос-платформні середовища виконання. Крос-платформне програмне забезпечення.

Змістовий модуль 2. Засоби розробки кросплатформних систем

Тема 1. Крос-платформні мови програмування.

Тема 2. Особливості кросплатформної мови програмування Ruby. Основні поняття. Особливості синтаксису. Переваги та недоліки.

Тема 3. Використання Ruby для Веб. Розгортання Ruby on Rails.

Тема 4. Особливості Ruby on Rails. Розробка веб-систем.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усь ого	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Модуль 1												
Змістовий модуль 1. Вступ до кросплатформного програмування												
Тема 1. Поняття про кросплатформність. Основні переваги та недоліки кросплатформного програмного забезпечення	20	2				18	20					20
Тема 2. Крос-платформність на рівні апаратної платформи. Крос-платформність на рівні ОС.	20	2		6		12	20					20
Тема 3. Крос-платформні середовища виконання. Крос-платформне програмне забезпечення	20	2		6		12	20					20
Разом за змістовим модулем 1	60	6		12		42	60					60
Модуль 2												
Змістовий модуль 2. Засоби розробки кросплатформних систем												
Тема 1. Крос-платформні мови програмування.	15	4		4		7	15	2				13
Тема 2. Особливості кросплатформної мови програмування Ruby. Основні поняття. Особливості синтаксису. Переваги та недоліки.	15	2		6		7	15	2		6		7
Тема 3. Використання Ruby для Веб. Розгортання Ruby on Rails.	15			4		11	15					15
Тема 4. Особливості Ruby on Rails. Розробка веб-систем.	15					16	15					15

Разом за змістовим модулем 2	60	6		14		40	60					50
Усього годин	120	12		26		82	120	4		6		110

5. Теми семінарських занять

№ з/п	Назва теми	Кількість годин
—	—	—

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
—	—	—

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Знайомство з мовою програмування Ruby	4
2	Робота з масивами в мові Ruby	4
3	Об'єктно-орієнтоване програмування у Ruby	6
4	Жадібні алгоритми в Ruby	6
5	Створення простого веб-додатку із використанням Ruby on Rails	4
6	Розробка бази даних із використанням Ruby on Rails	4
	Разом	28

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Основні поняття кросплатформного програмного забезпечення	18
2	Крос-платформеність на рівні апаратної платформи. Крос-платформність на рівні ОС.	12
3	Крос- платформні середовища виконання.	12
4	Крос- платформні мови програмування.	7
5	Особливості кросплатформної мови програмування Ruby. Основні поняття. Особливості синтаксису. Переваги та недоліки.	7
6	Використання Ruby для Веб. Розгортання Ruby on Rails.	11
	Разом	82

9. Індивідуальні завдання

Індивідуальні завдання навчальним планом не передбачено.

10. Методи навчання

В якості методів навчання використовуються: лекції, лабораторні роботи, консультації.

11. Методи контролю

Для контролю проводяться самостійні роботи, тестування, перевірка пройденого матеріалу шляхом опитування при здачі лабораторних робіт.

12. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота							Підсумковий тест (екзамен)	Сума
Змістовий модуль 1			Змістовий модуль 2				60	100
T1	T2	T3	T1	T2	T3	T4		
5	5	6	6	6	6	6		

T1, T2, T3, T4 – теми змістових модулів.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
85-89	B	добре	
75-84	C		
70-74	D	задовільно	
60-69	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

13. Методичне забезпечення

1. Методичні вказівки до виконання лабораторних робіт з дисципліни “Крос-платформне програмування” для студентів спеціальностей 121 Інженерія програмного забезпечення, 122 Комп'ютерні науки (всіх форм навчання) / А. О. Олійник, О. О. Олійник, Є. М. Федорченко, О. І. Качан. – Запоріжжя: ЗНТУ, 2016. – 19 с..

14. Рекомендована література

Базова література

1. Martin Fowler. Patterns of Enterprise Application Architecture. Addison Wesley Longman, Reading, MA, 2003.
2. Фаулер. Ч. Rails. Сборник рецептов. Питер, СПб, 2008.
3. Хант Э., Томас Д. Программист-прагматик. Питер, СПб, 2007.
4. Dion Almaer, Justin Gehtland, Ben Galbraith. Pragmatic AJAX: A Web 2.0 Primer. The Pragmatic Programmers, LLC, Raleigh, NC, Dallas, TX, 2006.
5. David Thomas, Chad Fowler, Andrew Hunt. Programming Ruby: The Pragmatic Programmers' Guide. The Pragmatic Programmers, LLC, Raleigh, NC, Dallas, TX, 2nd edition, 2005.
6. Томас Д., Хэнссон Д.Х. Гибкая разработка веб-приложений в среде Rails. – СПб. : Питер, 2008. – 716 с.

Додаткова література

1. Joe Armstrong. Programming Erlang: Software for a Concurrent World. The Pragmatic Programmers, LLC, Raleigh, NC, and Dallas, TX, 2007.
2. Martin Odersky, Lex Spoon, and Bill Venners. Programming in Scala. 2008.
3. David Thomas, Chad Fowler, and Andrew Hunt. Programming Ruby: The Pragmatic Programmers' Guide. The Pragmatic Programmers, LLC, Raleigh, NC, and Dallas, TX, third edition, 2008.
4. Guide to the Software Engineering Body of Knowledge (SWEBOK). – New York : IEEE Publising House, 2004. – 129 p.
5. David Thomas and David Heinemeier Hansson. Agile Web Development with Rails. The Pragmatic Programmers, LLC, Raleigh, NC, and Dallas, TX, 2005.
6. Simon Thompson. Haskell: The Craft of Functional Programming. Addison-Wesley, Reading, MA, second edition, 1999.