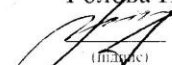


Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Керівник проєктної групи/
Голова НМК

 І.Б. Туркін
(ініціали) (ініціали та прізвище)

«30» 08 2019 р.

**РОБОЧА ПРОГРАМА ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Конструювання програмного забезпечення
(назва навчальної дисципліни)

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

Спеціальність: 121 «Інженерія програмного забезпечення»
(код і найменування спеціальності)

Освітня програма: «Інженерія програмного забезпечення»
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2019 рік

Робоча «Конструювання програмного забезпечення» для студентів за спеціальністю: 121 «Інженерія програмного забезпечення» освітньою програмою «Інженерія програмного забезпечення»

«20» 04 2019 р, – 10 с.

Розробник : Волобуєва Л.О., доц., к.т.н.

(прізвище та ініціали, посада, науковий ступінь і вчене звання)



(підпис)

Робочу програму розглянуто на засіданні кафедри інженерії програмного забезпечення (№ 603)

(назва кафедри)

Протокол № 1 від «30» 08 2019 р.

Завідувач кафедри д-р техн. наук., проф.

(науковий ступінь і вчене звання)



(підпис)

І.Б. Туркін

(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 4,5	<p style="text-align: center;">Галузь знань <u>12 «Інформаційні технології»</u> <small>(шифр і найменування)</small></p> <p style="text-align: center;">Спеціальність <u>121 «Інженерія програмного забезпечення»</u> <small>(код і найменування)</small></p> <p style="text-align: center;">Освітня програма <u>«Інженерія програмного забезпечення»</u> <small>(найменування)</small></p> <p style="text-align: center;">Рівень вищої освіти: перший (бакалаврський)</p>	Цикл професійної підготовки (1.2. Дисципліни загально-професійної підготовки)
Кількість модулів – 2		Навчальний рік
Кількість змістовних модулів – 2		2019/2020
Індивідуальне завдання _____ <small>(назва)</small>		Семестр
Загальна кількість годин – 64/135		5-й
	Лекції*	
	32 години	
	Практичні, семінарські* _____ годин	
	Лабораторні* 32 години	
	Самостійна робота 71 година	
	Вид контролю модульний контроль, іспит	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: **64/71**.

* Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: надання знань з математичних основ комп'ютерного та загальних принципів дисциплінованого детального проектування програмного забезпечення та формальних моделей, що реалізують ці принципи.

Завдання: опанування практичними навичками детального проектування програмного забезпечення

Результати навчання: студент має:

знати :

- формальний апарат для опису алгоритмічних мов (системи регулярних виразів, контекстно-вільні граматики, кінцеві автомати без пам'яті і із стековою пам'яттю);
- властивості формальних систем їх класифікацію і методи еквівалентних перетворень усередині своїх класів;
- методи перетворення формальних описів лексики і синтаксису мови в управляючі таблиці детермінованих оптимальних кінцевих автоматів;
- алгоритми лексичного, синтаксичного і семантичного аналізу, що реалізуються відповідними кінцевими автоматами;
- особливості проміжних форм представлення трансльованої програми;
- методи генерації об'єктного коду для конкретної цільової машини;
- методи оптимізації трансльованої програми;

вміти :

- використовувати широкий вибір методів і засобів розробки програмного забезпечення, включаючи методи, засновані на станах, і засновані на табличному описі підходи до детального проектування програмного забезпечення;
- проектувати прості мови і протоколи, що підходять для різних програм;
- генерувати код для простих мов і протоколів, використовуючи відповідні засоби;
- створювати прості формальні специфікації низькорівневих модулів, перевіряти правильність цих специфікацій і генерувати для них код, використовуючи відповідні засоби;
- проектувати просте паралельне програмне забезпечення;
- аналізувати програмне забезпечення для поліпшення його продуктивності, надійності і супроводжуємі.

мати уявлення:

- про хеш-функцію та хеш-адресацію;
- про динамічну організацію пам'яті ПК.

Міждисциплінарні зв'язки: Об'єктно-орієнтоване програмування.

3. Програма навчальної дисципліни

Модуль 1.

Змістовний модуль 1. Основи теорії граматики та синтаксичного аналізу

Тема 1. Формальні мови та граматики. Основні терміни та визначення. Дерево виведення. Задача розбору. Граматики Хомського. Ієрархія граматики Хомського. Контекстно-вільні граматики та мови. Однозначність КВ-граматики. Алгоритми розпізнання КВ-мов. Синтаксичні діаграми КВ-мов. LL-граматики. Автоматні граматики та мови. Граф автоматної граматики.

Тема 2. Скінченні автомати. Програмна реалізація автоматного розпізнавача. Дерево розбору в автоматній граматиці. Синтаксичні діаграми автоматної мови. Автомат розпізнання для КВ-мов.

Тема 3 Регулярні вирази та регулярні множини. Еквівалентність регулярних виразів та автоматних граматики. Синтаксичний аналіз арифметичних виразів. Включення дій в синтаксис. Обробка помилок при трансляції. Табличний LL(1)-аналізатор. Трансляція виразів. Інтерпретація виразів. Семантичне дерево виразу.

Модульний контроль

Модуль 2.

Змістовний модуль 2. Трансляція мов програмування. Методи управління паралелізмом та комунікацією між процесами

Тема 1. Опис мов програмування. Метамови. Використання форми Бекуса-Наура для опису мови програмування. Синтаксичні діаграми. Розширена форма Бекуса-Наура. Компілятор, інтерпретатор, конвертор. Структура компілятора. Багатопрохідні та однопрохідні транслятори. Лексичний аналізатор (сканер). Види та значення лексем. Синтаксичний аналізатор. Контекстний аналіз. Таблиця імен. Контекстний аналіз модуля. Трансляція списку імпорту. Трансляція описів. Контекстний аналіз виразів. Контекстний аналіз операторів.

Тема 2. Генерація коду. Віртуальна машина. Архітектура віртуальної машини. Програмування в коді віртуальної машини. Реалізація віртуальної машини. Генератор коду. Розподілення пам'яті. Генерація коду для виразів. Генерація коду для операторів. Завершення генерації. Призначення адрес змінним. Трансляція процедур. Розширений набір команд віртуальної машини. Особливості трансляції параметрів-змінних. Конструкція простого асемблера. Язык асемблера віртуальної машини. Реалізація асемблера. Автоматизація побудови та мобільність трансляторів. Автоматичний аналіз та перетворення граматики. Автоматична побудова компілятора та його частин. Використання мов високого рівня.

Тема 3 Методи управління паралелізмом та комунікацією між процесами. Методи проектування програмного забезпечення для чисельних розрахунків. Вступ до проміжного програмного забезпечення (middleware). Пошук критичних за часом ділянок та оптимізація продуктивності ПЗ.

Модульний контроль

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	усього	у тому числі			
		Л	П	лаб	с.р.
1	2	3	4	5	6
Змістовий модуль 1. Основи теорії граматики та синтаксичного аналізу					
Тема 1. Вступ. Формальні мови та граматики. Основні терміни та визначення. Граматики Хомського. Ієрархія граматики Хомського.	15	4		4	7
Тема 2. Контекстно-вільні граматики та мови.	16	4		4	8
Тема 3. Автоматні граматики та мови. Регулярні вирази та регулярні множини.	16	4		4	8
Тема 4. Трансляція виразів. Інтерпретація виразів. Семантичне дерево виразу.	12	4		4	4
Модульний контроль	4				4
Разом за змістовим модулем 1	63	16		16	31
Змістовий модуль 2. Трансляція мов програмування. Методи управління паралелізмом та комунікацією між процесами					
Тема 5. Опис мов програмування. Метамови. Використання БНФ для опису мови програмування.	16	4		4	8
Тема 6. Компілятор, інтерпретатор, конвертор. Структура компілятора. Багатопрхідні та однопрхідні транслятори.	14	2		4	8
Тема 7. Лексичний аналізатор (сканер). Види та значення лексем. Синтаксичний аналізатор. Контекстний аналіз.	16	4		4	8
Тема 8. Хеш-функція та хеш-адресація. Методи подолання колізій. Метод рехешування. Метод ланцюжків.	12	4			8
Тема 9. Генерація коду. Віртуальна машина. Трансляція процедур. Триадресний та чотириадресний код. Методи управління паралелізмом та комунікацією між процесами.	10	2		4	4
Модульний контроль	4				4
Разом за змістовим модулем 2	68	16		16	40
Усього годин	131	32		32	71

5. Теми семінарських занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

6. Теми практичних занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

7. Теми лабораторних занять

№ п/п	Назва теми	Кількість годин
1	ЛР 1. Побудова синтаксичної діаграми заданих конструкцій.	6
2	ЛР 2. Програмування синтаксичного аналізатора методом рекурсивного спуску.	6
3	ЛР 3. Реалізація табличного LL(1) аналізатора.	6
4	ЛР 4. Контекстна обробка виразів.	6
5	ЛР 5. Оптимізація коду за допомогою лексичної згортки тексту програми	8
	Разом	32

8. Самостійна робота

№ п/п	Назва теми	Кількість годин
1	Генеалогія мов програмування	7
2	Автоматичний аналіз та перетворення граматик	8
3	Автоматична побудова компілятора та його частин	8
4	Використання мов високого рівня	4
5	Розробка та застосування самокомпіляторів	8
6	Розширений набір команд віртуальної машини	8
7	Порівняння компіляторів різних систем програмування	8
8	Опис синтаксису найбільш поширених мов програмування	8
9	Порівняльна оцінка мов програмування. Генеалогія мов програмування	4

10	Модульний контроль	8
	Разом	71

9. Індивідуальні завдання

10. Методи навчання

Словесні: пояснення на лекційних та практичних заняттях, розповідь, бесіда.
Наочні: ілюстрування у вигляді слайдів презентації MS PowerPoint та рисунків MS Visio. Практичні: лабораторні роботи.

11. Методи контролю

Поточний контроль (під час проведення лабораторних занять) у формі лабораторних та розрахункових робіт.

Тестовий контроль (під час лекційних занять) у формі модульних контрольних робіт.

Семестровий контроль у формі письмового іспиту.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях	0...1	8	0...5
Виконання і захист лабораторних (практичних) робіт	6...10	3	18...30
Модульний контроль	0...20	1	0...20
Змістовний модуль 2			
Робота на лекціях	0...1	8	0...5
Виконання і захист лабораторних (практичних) робіт	6...10	2	12...20
Модульний контроль	0...20	1	0...20
Усього за семестр			60...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з чотирьох запитань (теоретичних та практичних), максимальна кількість балів за кожне питання 25 балів.

12.2. Якісні критерії оцінювання

Задовільно (60-74). Мати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи. Знати основи побудови формальних мов та граматики. Вміти створювати прості формальні специфікації низькорівневих модулів, перевіряти правильність цих специфікацій і генерувати для них код, використовуючи відповідні засоби.

Добре (75 - 89). Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Вміти пояснювати використовувати широкий вибір методів і засобів розробки програмного забезпечення, включаючи методи, засновані на станах, і засновані на табличному описі підходи до детального проектування програмного забезпечення, аналізувати програмне забезпечення для поліпшення його характеристик.

Відмінно (90 - 100). Повно знати основний та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Досконально знати формальний апарат для опису алгоритмічних мов (системи регулярних виразів, контекстно-вільні граматики, кінцеві автомати без пам'яті і із стековою пам'яттю) та методи трансляції. Вміти аналізувати та вдосконалювати програмне забезпечення для поліпшення його продуктивності, надійності і супроводжуємості. Безпомилково виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з докладним обґрунтуванням рішень та заходів, які запропоновано у роботах.

Розподіл балів, які отримують студенти за виконання курсової роботи (проекту) – *не передбачено навчальним планом*

Пояснювальна записка	Ілюстративна частина	Захист роботи	Сума
до ____	до ____	до ____	100

Шкала оцінювання: бальна і традиційна

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

13. Методичне забезпечення

Волобуева Л.А. Конструирование программного обеспечения : учеб. пособие по лаб. практикуму / Л. А. Волобуева, Т. Г. Дегтярева, В. А. Постернакова ; М-во образования и науки Украины, Нац. аэрокосм. ун-т им. Н. Е. Жуковского "Харьк. авиац. ин-т". - Харьков : Нац. аэрокосм. ун-т им. Н. Е. Жуковского "Харьк. авиац. ин-т", 2015. - 41 с .

Електронний ресурс, на якому розміщено навчально-методичний комплекс дисципліни: *mentor.khai.edu*

14. Рекомендована література

Базова

1 Волобуева Л.А. Конструирование программного обеспечения : учеб. пособие по лаб. практикуму / Л. А. Волобуева, Т. Г. Дегтярева, В. А. Постернакова ; М-во образования и науки Украины, Нац. аэрокосм. ун-т им. Н. Е. Жуковского "Харьк. авиац. ин-т". - Харьков : Нац. аэрокосм. ун-т им. Н. Е. Жуковского "Харьк. авиац. ин-т", 2015. - 41 с .

2 Конструювання програмного забезпечення. Лабораторний практикум./Х.: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2009./ Туркін І.Б., Михайлова Н.А., Волобуєва Л.О.

3 Свердлов С.З. Языки программирования и методы трансляции.: Учеб. пособ. - СПб.: Питер, 2007. – 638 с.

4 Пратт Т., Зелковиц М. Языки программирования: разработка и реализация. : Пер. с англ. / Под общей ред. А. Матросова. - СПб.: Питер, 2002. - 688 с.

5 Себеста Роберт У. Основные концепции языков программирования, 5-е изд. : Пер. с англ. - М.: Издательский дом "Вильямс", 2001. - 672 с.

6 Карпов Ю.Г. Теория и технология программирования. Основы построения трансляторов. - СПб.: БХВ-Петербург, 2005. - 272 с.

7 Хантер Р. Основные концепции компиляторов. : Пер. с англ. - М.: Издательский дом "Вильямс", 2002. - 256 с.

Орлов С.А. Теория и практика языков программирования. Учебник для вузов. 2-е изд. Стандарт 3-го поколения: Учебник для вузов. СПб.: Питер, 2017. – 688 с.

Допоміжна

1 Ахо А.В., Сети Р., Ульман Д.Д. Компиляторы: принципы, технологии и инструменты. - М.: Издательский дом "Вильямс", 2003. - 768 с.

2 Хопкрофт Дж.Э., Мотвани Р., Ульман Д.Д. Введение в теорию автоматов, языков и вычислений, 2-е изд. - М.: Издательский дом "Вильямс", 2002. - 528 с.

3 Фридли Дж. Регулярные выражения, 3-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2008. – 608 с., ил.

15. Інформаційні ресурси