

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук

“ЗАТВЕРДЖУЮ”
Декан факультету
інформаційних технологій
Глазунова О.Г.
“ 20 ” 06 2019р.

РОЗГЛЯНУТО І СХВАЛЕНО
на засіданні вченої ради
факультету інформаційних технологій
Протокол № 11 від “ 20 ” 06 2019р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ПРОГРАМУВАННЯ

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА

«Комп'ютерні науки»

за спеціальністю 122 «Комп'ютерні науки»

галузі знань 12 «Інформаційні технології»

Факультет інформаційних технологій

Розробники: завідувач кафедри комп'ютерних наук, доцент, к.т.н Голуб Б. Л.,
доцент кафедри комп'ютерних наук, с.н.с., к.т.н. Боярінова Ю.Є.

Київ 2019

1. Опис навчальної дисципліни

Програмування

Галузь знань, спеціальність, освітній ступінь			
Галузь знань	12 Інформаційні технології		
Спеціальність	122 «Комп’ютерні науки»		
Освітній ступінь	Бакалавр		
Характеристика навчальної дисципліни			
Вид	Обов’язкова		
	1 семестр	2 семестр	
Загальна кількість годин	90 год.	90 год.	
Кількість кредитів ECTS	3	3	
Кількість змістових модулів	2	2	
Курсовий проект (робота)	-		
Форма контролю	Залік	Іспит	
Показники навчальної дисципліни для денної форми навчання			
	денна форма навчання		заочна форма навчання
	1 семестр	2 семестр	
Рік підготовки	1 курс	1 курс	
Семестр	1	2	
Лекційні заняття	30 год.	30 год.	12 год.
Лабораторні заняття	30 год.	30 год.	14 год.
Самостійна робота	30год.	30 год.	154 год.
Курсова робота	0	0	
Кількість тижневих годин для денної форми навчання: аудиторних самостійної роботи студента –	4 год. 4 год.		

Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Програмування» є отримання студентами знань з області розробки алгоритмів та програмування. Оволодіння такими знаннями дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами, в тому числі, сільськогосподарськими, за допомогою комп'ютерної техніки. Такі знання майбутній спеціаліст зможе застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.

Вивчення дисципліни «Програмування» сприяє формуванню у студентів наступних компетентностей.

Загальні компетентності:

ЗК2. Здатність застосовувати знання у практичних ситуаціях;

ЗК6. Здатність вчитися й оволодівати сучасними знаннями;

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові, предметні) компетентності:

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, 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.

СК8. 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.

Це забезпечує досягнення програмних результатів навчання ПР1, ПР5, ПР9, згідно з якими студент повинен знати:

- визначення алгоритмів;
- технології розробки алгоритмів;

- використання алгоритмічних мов для реалізації розроблених алгоритмів на персональних комп'ютерах (ПК);
- технології розробки програм на мові С;
- базових конструкцій мови С;

оволодіти:

- практичними навичками розробки алгоритмів лінійної, розгалуженої та циклічної структур;
- умінням розроблювати програми на мові С, С++;
- практичними навичками реалізації алгоритмів на мові С, С++ та налагодження програм;
- умінням самостійно опановувати нові методи та технології розробки програм.

Викладання дисципліни ґрунтується на знаннях по таких напрямках як:

- іноземна мова (бажано, англійська);
- вища математика;
- архітектура комп'ютера;

Дисципліна “ Об’єктно-орієнтоване програмування ” повинна передувати дисциплінам:

- «Об’єктно-орієнтоване програмування»;
- «Системи і технології управління базами даних»;
- «Технологія програмування та створення програмних продуктів.

2. Програма навчальної дисципліни

Семестр I

Модуль №1. 3.01.01- Організація програм

Тема №1. Введення в програмування (2 години)

Алгоритми та програми. Мови програмування. Блок-схема як засіб графічного зображення алгоритмів. Лінійні алгоритми. Алгоритми з розгалуженням. Цикли та циклічні структури.

Тема №2. Основні поняття мови C (4 години)

Історія мови C. Характеристика C-систем.

Загальна структура програми на мові C. Директиви передпроцесора. Стандартні функції введення/виведення мови C. Використання коментарів.

Об'ява змінних. Загальні поняття щодо типів даних мови C. Числові типи та символьний тип.

Загальні поняття щодо пам'яті комп'ютера. Руйнуюче та не руйнуюче зчитування. Арифметика в мові C. Пріоритет виконання операцій.

Прийняття рішення: операції рівності та відношення. Пріоритет виконання розглянутих операцій. Ключові слова.

Тема №3. Оператори мови C (8 годин)

Операції присвоєння. Повна та скорочена форма операції присвоєння. Операції інкремента та декремента.

Структури управління. Структура вибору *if*. Структура вибору *if/else*.

Структура із множинним вибором *switch*. Приклади програм з реалізованими структурами управління.

Структури повторення. Структура повторення *for*. Блок-схема алгоритму структури *for*. Синтаксис структури *for*. Приклади структур *for*.

Структура повторення **while**. Структура повторення **do/while**. Рекомендації щодо вибору структур повторення в залежності від алгоритму. Приклади програм з реалізованими структурами повторення.

Оператори **break i continue**. Логічні операції. Пріоритет виконання розглянутих операцій.

Тема №4. Проектування програм з використанням покажчиків (8 годин)

Ідеологія використання покажчиків в програмі на мові С. Об'ява та ініціалізація покажчиків. Приклади використання покажчиків.

Операції з покажчиками. Виведення значень покажчиків як адрес оперативної пам'яті. Виведення значення, на яке посилається покажчик. Вирази та арифметика з покажчиками.

Доступ до апаратних елементів комп'ютера за допомогою покажчиків. Приклади програм з використанням покажчиків.

Модуль №2. 3.01.02- Масиви і алгоритми обробки

Тема №5. Масиви (8 годин)

Масиви як структурований тип даних. Визначення масивів. Синтаксис об'яви масивів. Поняття “елемент масиву” та “індекс масиву”. Синтаксис використання масивів.

Зв'язок покажчиків з масивами. Вирази та арифметика з покажчиками. Динамічний розподіл пам'яті.

Стандартні алгоритми роботи з масивами: визначення екстремальних значень, статистичних показників, упорядкування масивів, пошук визначеного елемента.

Багатовимірні масиви. Приклади програм з багатовимірними масивами. Реалізація алгоритмів, пов'язаних з обробкою матриць.

Семестр II

Модуль №3. 3.01.03- Функції і управління пам'яттю

Тема №6. Використання функцій на мові С (10 годин)

Загальні поняття щодо структурного програмування. Програмні модулі мови С. Функції математичної бібліотеки. Функції бібліотеки стандартного введення/виведення.

Використання функцій при розробці програм на мові С. Визначення функцій. Прототипи функцій та файли заголовків.

Розробка власних функцій. Виклик функції за значенням. Способи обміну інформацією між функціями. Локальні та глобальні змінні.

Виклик функції за посиланням – передача параметрів за адресою. Приклад програми, що використовує виклик за посиланням.

Класи пам'яті. Правила області дій. Розробка програми у вигляді проекту. Зовнішні змінні.

Рекурсія. Приклад використання рекурсії: числа Фібоначчі.

Модуль №4. 3.01.04- Структури даних і робота з файлами

Тема №7. Використання структур для створення власних типів даних.

Використання об'єднань, операцій з бітами (10 годин)

Створення власних типів даних за допомогою структур. Об'ява структур. Об'ява структурних змінних. Доступ до елементів структури.

Використання масивів структур. Використання структур з функціями. Реалізація зв'язаних списків, стеків, черг та дерев за допомогою структур.

Створення власних типів даних за допомогою об'єднань. Об'ява об'єднань та змінних-об'єднань.

Бітові операції. Бітові поля.

Константи, що перераховуються.

Тема №8. Файлова система (10 годин)

Загальні уявлення про файлову систему. Ієрархія даних. Файли і потоки.

Стандартні методи для роботи з файлами на мові С.

Створення файлів послідовного доступу. Читання із файлів послідовного доступу.

Приклади програм.

Файли довільного доступу. Створення файлів довільного доступу. Приклади програм.

Довільний запис у файл довільного доступу. Послідовне читання даних із файлів довільного доступу.

4. Структура навчальної дисципліни

Назви змістових модулів і тем		Кількість годин												
		денна форма							Заочна форма					
		тижні	усього	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	14
	Семестр I													
	Змістовий модуль 1. Організація програм													
Тема 1. Введення в програмування	1	6	2	-	2	-	2							
Тема 2. Основні поняття мови C	2	12	4	-	4	-	4							
Тема 3. Оператори мови C	4	24	8	-	8	-	8							
Тема 4. Проектування програм з використанням покажчиків	4	24	8		8		8							
Разом за змістовим модулем 1	11	66	22	-	22	-	22							
	Змістовий модуль 2. Масиви і алгоритми обробки													
Тема 5. Масиви	4	24	8	-	8	-	8							
Разом за змістовим модулем 2	4	24	8	-	8	-	8							
Разом за семестр I	15	90	30	-	30	-	30							
	Семестр II													
	Змістовий модуль 3. Функції і управління пам'яттю													
Тема 6. Використання функцій на мові C	5	30	10	-	10	-	10							
Разом за змістовим модулем 3	5	30	10	-	10	-	10							
	Змістовий модуль 4. Структури даних і робота з файлами													
Тема №7. Використання структур. Використання	5	30	10	-	10	-	10							

об'єднань, операцій з бітами.													
Тема №8. Файлова система	5	30	10	-	10	-	10						
Разом за змістовим модулем 4	10	60	20	-	20	-	20						
Разом за семестр 2	15	90	30	-	30	-	30						
Усього годин	30	180	60		60		60	180	12		14		154

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
	Семестр I	
1	С-системи. Змінні та типи змінних, об'ява змінних. Виведення та введення даних. Оператор присвоєння.	2
2	Проста та скорочена форми операції присвоєння. Інкремент та декремент. Структура вибору <i>if, if/else</i> .	4
3	Структура прийняття рішення switch	4
4	Логічні операції	2
5	Структура повторення <i>for</i> .	4
6	Структура повторення while (повторення з передумовою або повторення, що керуються контрольним значенням).	4
7	Структура повторення do/while .	4
8	Показчики. Об'ява та ініціалізація змінної-показчика.	2
9	Масиви. Визначення масивів. Багатовимірні масиви.	4
	Семестр II	
10	Зв'язок між показчиками і масивами. Вирази та арифметичні операції з показчиками. Динамічний розподіл пам'яті.	4
11	Використання функцій. Визначення власних функцій власних функцій, які відповідають окремим частинам загального алгоритму.	4
12	Класи пам'яті. Правила області дії	4
13	Робота з символами	4
14	Обробка символьних рядків	4
15	Опанування практичними навичками розробки програм із використанням структур. Навчитися використовувати операції для доступу до елементів структури.	4

16	Об'єднання. Операції з бітами.	2
17	Робота з файлами. Створення та читання файла послідовного доступу.	2
18	Робота з файлами. Створення та читання файла довільного доступу.	2
	Всього	60

6. Самостійна робота

Семестр I

1. Порівняльна характеристика мов програмування – 2 год.
2. Методології розробки програм: низхідне та висхідне проектування – 2 год.
3. Виведення даних із визначеною шириною поля і точністю представлення. Використання прапорців в рядку управління форматів функції **printf**. Друк літералів та Esc-послідовностей – 2 год.
4. Стандартні бібліотеки середовища Borland C++ – 2 год.
5. Показники як спосіб низько рівневого програмування – 3 год.
6. Алгоритми сортування великих масивів - 4 год.

Семестр II

7. Завершення роботи програми за допомогою макросів **exit**, **atexit** – 2 год.
8. Основні директиви передпроцесора мови C. Використання директив **#include**, **#define**, **#ifdef** – 2 год.
9. Додаткові директиви передпроцесора **#error**, **#pragma**. Операції #, ## - 2 год.
10. Структури, які посилаються самі на себе. Зв'язані списки, черги, дерева – 3 год.
11. Рекурсія – переваги і недоліки – 2 год.
12. Низько рівневий доступ до файлів – 4 год.

8. Методи навчання

При викладанні дисципліни використовуються наступні методи навчання:

М1. Лекція (проблемна, інтерактивна)

М2. Лабораторна робота – для використання набутих знань до розв'язування практичних завдань;

9. Форми контролю

При викладанні дисципліни передбачені такі форми контролю:

МК1. Тестування

МК2. Контрольне завдання

МК4. Методи усного контролю

МК5. Екзамен

МК6. Залік

МК7. Звіт

Для студентів денної форми навчання: усне опитування (МК4) та експрес контроль (МК1) на лабораторних заняттях, захист індивідуальних лабораторних завдань (МК7), аудиторні модульні контрольні роботи (МК2).

10. Розподіл балів, які отримують студенти

Поточний контроль				Рейтинг з навчальної роботи $R_{нр}$	Рейтинг з додаткової роботи $R_{др}$	Рейтинг штрафний $R_{штр}$	Підсумкова атестація (екзамен чи залік)	Загальна кількість балів
Змістовий модуль 1	Змістовий модуль 2	Змістовий модуль 3	Змістовий модуль 4					
0-100	0-100	0-100	0-100	0-70	0-20	0-5	0-30	0-100

Примітки. 1. Відповідно до «Положення про кредитно-модульну систему навчання в НУБіП України», затвердженого ректором університету 03.04.2009 р., рейтинг студента з навчальної роботи $R_{нр}$ стосовно вивчення певної дисципліни визначається за формулою

$$0,7 \cdot (R^{(1)}_{зм} \cdot K^{(1)}_{зм} + \dots + R^{(n)}_{зм} \cdot K^{(n)}_{зм})$$

$$R_{нр} = \frac{\dots}{K_{дис}} + R_{др} - R_{штр},$$

де $R^{(1)}_{зм}, \dots, R^{(n)}_{зм}$ – рейтингові оцінки змістових модулів за 100-бальною шкалою;

n – кількість змістових модулів;

$K^{(1)}_{зм}, \dots, K^{(n)}_{зм}$ – кількість кредитів ECTS, передбачених робочим навчальним планом для відповідного змістового модуля;

$K_{дис} = K^{(1)}_{зм} + \dots + K^{(n)}_{зм}$ – кількість кредитів ECTS, передбачених робочим навчальним планом для дисципліни у поточному семестрі;

$R_{др}$ – рейтинг з додаткової роботи;

$R_{штр}$ – рейтинг штрафний.

Наведену формулу можна спростити, якщо прийняти $K^{(1)}_{зм} = \dots = K^{(n)}_{зм}$. Тоді вона буде мати вигляд

$$R_{нр} = \frac{0,7 \cdot (R^{(1)}_{зм} + \dots + R^{(n)}_{зм})}{n} + R_{др} - R_{штр}.$$

Рейтинг з додаткової роботи $R_{др}$ додається до $R_{нр}$ і не може перевищувати 20 балів. Він визначається лектором і надається студентам рішенням кафедри за виконання робіт, які не передбачені навчальним планом, але сприяють підвищенню рівня знань студентів з дисципліни.

Рейтинг штрафний $R_{штр}$ не перевищує 5 балів і віднімається від $R_{нр}$. Він визначається лектором і вводиться рішенням кафедри для студентів, які матеріал змістового модуля засвоїли невчасно, не дотримувалися графіка роботи, пропускали заняття тощо.

2. Згідно із зазначеним Положенням *підготовка і захист курсового проекту (роботи)* оцінюється за 100 бальною шкалою і далі переводиться в оцінки за національною шкалою та шкалою ECTS.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою	
	для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	відмінно	зараховано
89-74	добре	
73-60	задовільно	
35-59	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

11. Методичне забезпечення

1. Б.Л.Голуб \ Б.Л.Голуб, Є.М.Шукайло. Методичний посібник до вивчення дисципліни “Програмування та алгоритмічні мови”. Методичний посібник. – Видавничий центр НУБіП, 2012. – 64 с.
2. Б.Л.Голуб \ Б.Л.Голуб, О.І.Примак. Методичні вказавки до виконання лабораторних робіт з дисциплін “Основи програмування”, “Програмування та алгоритмічні мови” (частина І). – Видавничий центр НУБіП, 2009. – 38 с.
3. Б.Л.Голуб \ Б.Л.Голуб, О.І.Примак. Методичні вказавки до виконання лабораторних робіт з дисциплін “Основи програмування”, “Програмування та алгоритмічні мови” (частина ІІ). – Видавничий центр НУБіП, 2009. – 50 с.

12. Рекомендована література

Базова

1. Б.Керниган \ Б.Керниган, Д.Ритчи. Язык программирования С. – Санкт-Петербург, 2001. – 300 с.
2. Х.М.Дейтел \ Х.М.Дейтел, П.Дж.Дейтел. Как программировать на С. – М.:”Бином”, 2000. – 1005 с.

Додаткова

3. Крис Паппас \ Крис Паппас, Уильям Мюррей. Программирование на С и С++. Серия «Библиотека студента». – «Ирина», ВНУ, Киев, 2000. – 320с.
4. ANSI, American National Standard for Information Systems – Programming Language C. – New York, 1990.

13. Інформаційні ресурси

1. ЕНК по цій дисципліні знаходиться за електронною адресою:
<http://it.nubip.edu.ua/course/view.php?id=91>

Інтернет курси

<https://www.intuit.ru/studies/courses/97/97/info>

<http://edu-cisco.org/#wpcf7-f6353-p2-o1>

<https://ravesli.com/uroki-cpp/>

Інтернет ресурси

[https://uk.wikipedia.org/wiki/C_\(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F\)](https://uk.wikipedia.org/wiki/C_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))

<http://www.c-cpp.ru/>

<http://cppstudio.com/cat/271/273/>

<https://www.e-olymp.com/ru/blogs/posts/28>

<https://intellect.icu/7-primery-na-yazyke-si-literatura-4309>