

Міністерство освіти і науки України

**Державний вищий навчальний заклад
«Донбаський державний педагогічний університет»**

Кафедра методики навчання математики та методики навчання інформатики

НАВЧАЛЬНА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Технології програмування

(назва навчальної дисципліни)

**підготовки здобувачів ступеня вищої
освіти**

бакалавр

(назва рівня вищої освіти)

спеціальності 014.04 Середня освіта (математика)

(шифр і назва напрямку підготовки)

спеціалізації

(назва спеціалізації)

Слов'янськ – 2017 р.

РОЗРОБЛЕНО ТА ВНЕСЕНО КАФЕДРОЮ МЕТОДИКИ НАВЧАННЯ
МАТЕМАТИКИ ТА МЕТОДИКИ НАВЧАННЯ ІНФОРМАТИКИ ФІЗИКО-
МАТЕМАТИЧНОГО ФАКУЛЬТЕТУ ДВНЗ «ДДПУ»

УКЛАДАЧ ПРОГРАМИ:

Величко В.Є. кандидат фізико-математичних наук, доцент, доцент кафедри
методики навчання математики та методики навчання інформатики ДВНЗ «ДДПУ»

РЕЦЕНЗЕНТИ:

Ровенська О.Г. - кандидат фізико-математичних наук, доцент, доцент кафедри
вищої математики ДДМА

Новіков О.О. – кандидат фізико-математичних наук, доцент, доцент кафедри
математики та інформатики ДДПУ

Рекомендовано до впровадження
науково-методичною радою
Державного вищого навчального закладу
«Донбаський державний педагогічний університет»

«21» вересня 2017 р.
протокол № 2

Перший проректор _____ Набока О.Г.

ВСТУП

Навчальна програма дисципліни «Технології програмування» складена відповідно до освітньої програми та навчального плану підготовки здобувачів ступеня вищої освіти «бакалавр» за напрямом підготовки (спеціальністю) 014.04 Середня освіта (математика)

Предметом вивчення навчальної дисципліни є технології програмування.

Міждисциплінарні зв'язки: дискретна математика, інформатика, математична логіка та теорія алгоритмів.

Програма навчальної дисципліни містить такі *змістові модулі*:

1. Процедурно-орієнтоване програмування.
2. Об'єктно-орієнтоване програмування.

1. Мета й завдання навчальної дисципліни

1.1. Метою вивчення навчальної дисципліни «Технології програмування» є надання студенту теоретично обґрунтованих знань та наочно сформованих умінь використання технологій програмування для розв'язування складних алгоритмічних задач.

1.2. Основними завданнями вивчення навчальної дисципліни «Технології програмування» є формування базових знань з методів побудови алгоритмів, огляд сучасних засобів реалізації алгоритмів, формування навичок самоосвітньої діяльності із використанням інформаційних технологій.

1.3. За результатами вивчення дисципліни у здобувачів повинні бути сформовані такі компетентності:

загальні: здатність учитися, здатність до критики й самокритики, креативність, здатність до системного мислення, адаптивність і комунікабельність, здатність до творчості, наполегливість у досягненні мети, турбота про якість виконуваної роботи; навички роботи з комп'ютером, базові знання в галузі інформатики й сучасних інформаційних технологій;

спеціальні: навички створення алгоритмів і навички їх реалізації, уміння створювати програми, навички управління інформацією, володіння інформаційними технологіями, спроможність знаходити інформацію, систематизувати та узагальнювати її.

На вивчення навчальної дисципліни відведено 90 годин / 3 кредити ECTS.

2. Інформаційний обсяг навчальної дисципліни

Змістовий модуль 1.

Процедурно-орієнтоване програмування

Поняття інформації, інформаційні потоки. Процес обробки інформації за допомогою комп'ютерних технологій. Алгоритми та програми. Мови програмування. Блок-схема як засіб графічного зображення алгоритмів. Лінійні алгоритми. Алгоритми процесів з розгалуженнями та їх програмування. Алгоритми циклічних процесів: «до», «поки», «для». Регулярні та ітераційні цикли. Програмування циклів. Поняття даних. Організація даних. Подання чисельного типу даних, символного та логічного типу. Процедури та функції. Локальні та глобальні змінні. Формальні та фактичні змінні Рекурсія: пряма і непряма. Рекурсивні функції. Подвійна рекурсія. Поняття структурованої величини. Множини. Операції над множинами. Одновимірні масиви. Основні операції над масивами Розв'язання задач з кількісними характеристиками, створення та перетворення масивів. Упорядкування одновимірних масивів. Метод «бульбашки», вибору, включення, швидке сортування. Двовимірний масив. Особливості опису n-мірних масивів. Розв'язання задач на двовимірні масиви. Поняття запису. Поняття файлу. Файли прямого та послідовного доступу. Типізовані файли. Стандартні функції та процедури для роботи з типізованими файлами. Текстові файли. Нетипізовані файли. Поняття модульного програмування. Модулі користувача. Стандартні модулі. Використання оверлеїв при використанні процедур та модулів. Бібліотеки. Поняття динамічних змінних. Побудова списків. Побудова стеків. Використання стеків при складанні алгоритмів.

Змістовий модуль 2.

Об'єктно-орієнтоване програмування

Принцип інкапсуляції даних. Конструктори, деструктори. Класи і об'єкти, атрибути об'єктів: члени класів, функції класу, функції поза класом, область видимості і права доступу. Інтерфейс класу, реалізація класу. Визначення і оголошення класу. Вбудовані функції класу. Ідентичність об'єктів, копіювання, ініціалізація і присвоєння, ліквідація об'єктів. Поточний об'єкт self. Створення та ініціалізація об'єктів. Властивості конструкторів, довизначення конструкторів, замовчуваний конструктор, обмеження прав доступу до конструктора. Ліквідація об'єктів, властивості деструкторів. Довизначення (overloading) операторів (операцій-функцій): арифметичні операції (бінарні, унарні, суміщені з присвоєнням), оператори управління пам'яттю, оператор виклику функції, оператор доступу до елемента масиву, оператор доступу за указником, оператор присвоєння, конвертори (перетворення типів). Ієрархія об'єктів і ієрархія класів. Базовий клас, дочірній клас. Успадкування, уточнення привілеїв доступу до членів базового класу. Захищена область класу.

Область видимості і оператор розв'язування області видимості. Створення дочірніх об'єктів. Конструктор дочірнього класу, послідовність виклику конструкторів, ліквідація дочірніх об'єктів, деструктор дочірнього класу, послідовність виклику деструкторів. Статичне і динамічне зв'язування: поліморфізм, віртуальні функції, заміщення функцій, віртуальні деструктори.

3. Рекомендована література

1. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч. пос. — М.: Издательский дом “Вильямс”, 2000. — 384 с., ил.
2. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования. Харьков: Фолио; Ростов н/Д: Феникс, 1997. — 368 с.
3. Вентцель Е.С. Исследование операций: задачи, принципы, методология. — 2-е изд., стер. — М.: Наука. Гл. ред. физ.-мат. лит., 1988. — 208 с. — (Пробл. науки и техн. прогресса)
4. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. — М.: Мир, 1989. — 360 с., ил.
5. Караванова Т.П. Інформатика: методи побудови алгоритмів та їх аналіз: необчислювальні алгоритми: .: Навч. посіб. для 9-10 кл. із поглибл. вивч. інф-ки – К.: Генеза. – 2006.- 216 с
6. Караванова Т.П. Інформатика: основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. для 8-9 кл. із поглибл. вивч. інф-ки – К.: Генеза. – 2006.- 286 с.
7. Керман, Митчел, К. Программирование и отладка в Delphi. Учебный курс. Пер. с англ.. — М.: Издательский дом “Вильямс”, 2002, 672 с.: ил. – Парал. тит.англ.
8. Кнут Д. Искусство программирования. — М.: Вильямс, 2000
9. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2001. — 960 с., 263 ил.
10. Липский В. Комбинаторика для программистов: Пер. с польск. — М.: Мир, 1988. — 213 с., ил.

4. Форма підсумкового контролю результатів навчання

Залік

5. Засоби діагностики результатів навчання

Поточне оцінювання, захист виконаних лабораторних робіт, захист контрольних робіт, тестування.