

ІНСТИТУТ ЦИВІЛЬНОЇ АВІАЦІЇ

Кафедра інформаційних технологій

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОСНОВИ ПРОГРАМУВАННЯ

Рівень вищої освіти	перший (бакалаврський)
Ступінь вищої освіти	Бакалавр
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення

Шифр ОК15

**Х а р к і в
2 0 1 9**

Розроблено та внесено:

Інститутом цивільної авіації Харківського національного університету Повітряних Сил імені Івана Кожедуба.

Розробники програми: Федорченко Володимир Миколайович – доцент кафедри інформаційних технологій, кандидат технічних наук, доцент.

Ухвалено на засіданні вченої ради

Харківського національного університету Повітряних Сил
імені Івана Кожедуба

Протокол від "___" _____ 201_ року, № __

1. Загальна мета навчальної дисципліни

Метою викладання навчальної дисципліни "Основи програмування" є засвоєння необхідних знань щодо основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових типів даних, вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розроблення програм мовами C/C++.

Анотація: дисципліну "Основи програмування" віднесено до групи освітньо-професійних дисциплін підготовки бакалаврів за спеціальності 121 Інженерія програмного забезпечення.

Сьогоднішні умови господарювання вимагають від фахівців всебічного використання новітніх інформаційних технологій. Широкі можливості комп'ютеризованих засобів в питаннях збирання, оброблення та видавання необхідної інформації здатні значно підвищити якість розрахунків, зробити більш ефективним процес обґрунтування наукових, економічних, управлінських рішень.

Навчальна дисципліна є інструментальною основою для виконання аналітичної частини подальших спецкурсів, а також курсових і дипломних робіт. Вона забезпечує наступні дисципліни: "Об'єктно-орієнтоване програмування", "Алгоритми та структури даних", "Операційні системи та системне програмування", "Бази даних", "Розробка мобільних застосувань", "WEB-програмування", "Тестування програмного забезпечення", "Кросплатформене програмування", "Програмування для мобільних пристроїв".

3. Компетентності, які набуваються під час засвоєння навчальної дисципліни

1. ФК-3 Здатність розробляти архітектури, модулі та компоненти програмних систем.
2. ФК-13 Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

4. Запланований результат навчання

Згідно з вимогами освітньої програми визначені та сформульовані наступні результати навчання студентів:

1. РН-1 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
2. РН-4 Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

3. РН-6 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.
4. РН-7 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.
5. РН-13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.
6. РН-15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.
7. РН-17 Вміти застосовувати методи компонентної розробки програмного Забезпечення.

5. Структура навчальної дисципліни

5.1. Короткий зміст навчальної дисципліни

Блок змістових модулів 1. Лексичні основи мов C/C++.

Змістовий модуль 1.1. Етапи розроблення та впровадження програм.

Електронний числовий інтегратор ENIAC. Логічна організація універсального обчислювального пристрою. Відкрита архітектура системних блоків. Принципи Джона фон Неймана.

Класифікація систем числення. Особливості позиційних систем числення. Основа, алфавіт та базис позиційних систем числення. Подання чисел в позиційних системах числення. Переведення чисел з однієї системи числення в іншу.

Початкові відомості про технологію програмування: процедурне, структурне і об'єктно-орієнтоване програмування. Мови програмування: процедурні, аплікативні, системи правил, об'єктно-орієнтовані. Стандартизація мов та середовища проектування. Транслятори. Редактори. Компонувальники. Відладчики. Керівні структури: оператори, вирази та підпрограми. Огляд сучасних інтегрованих систем програмування. Інтегроване середовище системи програмування Visual C++.NET. Платформа DOT.NET. Етапи розроблення та впровадження програм. Вимоги до програмного коду.

Поняття алгоритму. Властивості алгоритму. Типові алгоритмічні конструкції. Розроблення алгоритму методом покрокового уточнення.

Змістовий модуль 1.2. Елементи алгоритмічних мов C/C++: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази.

Стандарти мов програмування C/C++. Структура C++ програми.

Лексичні елементи мови C++: алфавіт, коментарі, ідентифікатори, службові слова, дані, вираз, операнд, змінна, операція. Домовленості про імена.

Поняття типу даних. Класифікація і подання даних. Базові типи даних: логічний, символічний, цілий, речовинний. Перетворення типів: неявні перетворення, явні перетворення типу.

Пріоритети операцій. Таблиця пріоритетності й асоціативності операцій.

Операції. Унарні операції: унарний мінус, унарний плюс, порозрядне інвертування, логічне заперечення, інкремент, декремент, операція обчислення розміру (sizeof). Бінарні операції: адитивні, мультиплікативні, операції зсувів, порозрядні, операції відносин, логічні, привласнення, операція “кома”.

Стандартні математичні функції.

Константні величини: цілі, дійсні, перелічувальні, символічні (літерні), рядкові (рядки або літерні рядки). Правила визначення компілятором констант. Визначення констант за допомогою ключового слова const.

Змістовий модуль 1.3. Структурне програмування: послідовність, розгалуження та цикли.

Загальні відомості про систему вводу-виводу даних.

Вирази, символи пропусків, блоки і комплексні вирази. Операнд, змінна. Оператор привласнення. Оголошення й ініціалізація змінних.

Типи операторів. Найпростіший оператор, оператор-оголошення, оператор-визначення, оператор-вираз.

Керівні оператори: оператори проходження; оператори вибору (єдиний вибір – if, подвійний вибір – if/else, множинний вибір – switch, умовна операція); оператори повторення (оператор while, оператор do-while, оператор for). Вкладені цикли. Керівні оператори в циклах: оператор break, оператор continue, оператор goto. Рекомендації щодо вибору циклів.

Змістовий модуль 1.4. Передпроцесорна обробка.

Директиви препроцесора. Основи апарату макросів.

Директива препроцесора #include і файли, що включаються. Директива препроцесора #define: оголошення констант і макросів. Умовна компіляція. Використовування ключового слова typedef. Відмінність директиви #define від оператора typedef. Оператор typeid.

Змістовий модуль 1.5. Процедурно-орієнтоване програмування. Рекурсія.

Загальні відомості про функції. Структура функції.

Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції.

Локальні та глобальні змінні. Правило видимості змінних. Перетворення типів аргументів функцій. Правила автоматичного (неявного) перетворення типів. Явні перетворення типів. Правила роботи з функціями. Класи пам'яті.

Список параметрів функції. Параметри за умовчанням. Способи передачі параметрів. Способи повернення значення.

Функції, що підставляються.

Створення власних заголовних файлів.

Перевантаження функцій. Рекурсія. Типи рекурсії.

Робота функцій. Розбиття пам'яті. Стек і функції.

Модифікатори функцій.

Змістовий модуль 1.6. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування.

Методологія розроблення ПЗ: каскадне розроблення, інтерактивне розроблення.

Проектування програм, принцип модульності.

Методи проектування програм: низхідне та висхідне проектування, метод розширення ядра.

Блок змістових модулів 2. Основи програмування на мовах C/C++.

Змістовий модуль 2.1. Масиви.

Оголошення масивів. Ініціалізація масивів. Обробка одновимірних масивів даних.

Алгоритми сортування масивів.

Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Масиви і графи.

Масиви як параметри функцій.

Змістовий модуль 2.2. Похідні типи даних. Рядки в стилі C.

Рядки як масиви символів. Операції з рядками. Введення-виведення рядків.

Поняття покажчика, посилання. Покажчики і масиви. Адресна арифметика. Посилання. Приклади використання покажчиків і посилань. Покажчики на функції.

Параметри функцій як посилання.

Організація пам'яті в сучасних процесорах і покажчики мови C++. Моделі пам'яті. Статичні і динамічні змінні. Оператори new і delete. Динамічні масиви. Динамічні масиви як параметри функцій.

Змістовий модуль 2.3. Структури та об'єднання. Динамічні структури даних.

Структури, оголошення та визначення об'єктів типу структура. Відмінності структур мов C та C++. Методи в структурах. Статичні члени структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій. Структури як значення функції, що повертається.

Масиви структур. Покажчики на структури. Передача по посиланню масивів структур.

Об'єднання. Операції з об'єднаннями. Переліки. Змінні структури.

Визначення, класифікація динамічних структур даних, способи оголошення, ініціалізація динамічних структур, методи доступу до даних динамічних структур, розміщення їх у пам'яті, переваги і недоліки використання динамічних структур в програмах.

Зв'язні списки, однозв'язні і двонаправлені списки, циклічні списки. Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Створення двонаправленого списку. Прохід двонаправленого списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Змістовий модуль 2.4. Введення в систему вводу-виводу C/C++. Файлові структури даних.

Базові положення системи вводу-виводу C/C++. Потоки і буфери. Стандартні об'єкти вводу-виводу. Ієрархія класів вводу-виводу C++.

Ввід даних за допомогою глобального об'єкту cin.

Вивід рядків. Введення одного символу. Використовування функції get(): без параметрів, з параметрами. Ввід рядків із стандартного пристрою введення. Використовування функції getline().

Вивід даних за допомогою глобального об'єкту cout. Очищення буфера виведу – flush(). Використовування функцій put() і write(). Ввід – вивід даних, що форматується.

Маніпулятори вводу-виводу. Функції width(), precision(), fill(). Маніпулятори, визначувані користувачем.

Використовування файлів для вводу-виводу даних. Створення файлу. Створення потоку. Відкриття потоку. «Приєднання» файлу до потоку.

Режими відкриття та доступу до файлів. Обміни з файлом з допомогою потоку. «Від'єднання» потоку від файлу. Закриття файлу. Знищення файлу. Визначення стану потоку. Функції управління вводу-виводу.

Змістовий модуль 2.5. Шаблони. Стандартна бібліотека шаблонів.

Основи апарату шаблонів. Шаблони функцій. Шаблони структур. Перевантаження шаблонів функцій. Шаблони функцій сортування.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Алгоритми. Функціональні об'єкти. Ітератори, ітератори потоку. Робота з векторами, списками, стеками, чергами. Тип даних string. Зв'язок класу string з STL. Оголошення та визначення об'єктів типу string. Методи та властивості класу string. Операції з рядками. Функції роботи з рядками. Ввід-вивід рядків. Перетворення між string та рядками в стилі C.

Змістовий модуль 2.6. Обробка виключень.

Обробка виключень в мові C, функції abort(), exit(), atexit().

Принципи обробки виключень в мові C++. Генерація виключень. Послідовність обробки виключень. Особливості обробки виключень при динамічному виділенні пам'яті.

5.2. Схема проходження курсу

Схема проходження курсу відповідає робочій навчальній програмі навчальної дисципліни та розміщена в системі дистанційного навчання MOODLE. Система оцінювання сформованих компетентностей у студентів враховує види занять, які згідно з програмою навчальної дисципліни передбачають лекційні, лабораторні заняття, а також виконання самостійної роботи. Оцінювання сформованих компетентностей у студентів здійснюється за накопичувальною 100-бальною системою. Контрольні заходи включають:

поточний контроль, що здійснюється протягом семестру під час проведення лекційних, практичних, лабораторних занять і оцінюється сумою набраних балів (максимальна сума – 60 балів; мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів);

модульний контроль, що проводиться з урахуванням поточного контролю за відповідний змістовий модуль і має на меті *інтегровану* оцінку результатів навчання студента після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля;

підсумковий/семестровий контроль, що проводиться у формі семестрового екзамену, відповідно до графіка навчального процесу.

Поточний контроль з цієї навчальної дисципліни проводиться в таких формах:

- активна робота на лекційних заняттях;
- активна участь у виконанні завдань на практичну роботу;
- виконання та захист практичних робіт;
- захист індивідуального завдання;
- проведення поточного тестування;
- експрес-опитування.

Модульний контроль з цієї навчальної дисципліни проводиться у формі модульної письмової контрольної роботи.

Підсумковий/семестровий контроль проводиться у формі семестрового екзамену. **Семестрові екзамени** – форма оцінювання підсумкового засвоєння студентами теоретичного та практичного матеріалу з окремої навчальної дисципліни, що проводиться як контрольний захід.

6. Індивідуальні завдання

Робочою програмою навчальної дисципліни передбачено наступні індивідуальні завдання: розрахунково-графічна робота.

Індивідуальні навчально-дослідні завдання (ІНДЗ) виконуються самостійно при консультуванні викладачем на протязі вивчення дисципліни у відповідності до графіку навчального процесу за рахунок часу відведеного на самостійну роботу.

ІНДЗ виконуються з метою закріплення, поглиблення і узагальнення знань, одержаних студентами за час навчання та придбання практичних навичок їх застосування при вирішенні практичних задач; підготовки до виконання курсових і дипломних робіт.

Індивідуальне навчально-дослідне завдання припускає наявність наступних елементів наукового дослідження:

практичної значущості;

комплексного системного підходу до вирішення завдань дослідження;

теоретичного використання передової сучасної методології і наукових розробок;

наявність елементів творчості.

В процесі виконання ІНДЗ, разом з теоретичними знаннями і практичними навиками за фахом, студент повинен продемонструвати здібності до науково-дослідної роботи і уміння творчо мислити, навчитися вирішувати науково-прикладні актуальні задачі.

Тематика ІНДЗ:

«Обробка матриць».

Мета роботи: набуття практичних навичок розробки алгоритмів і програм обробки масивів даних

Основні завдання:

- 1) розробка графічної схеми алгоритму розв'язання практичної задачі;
- 2) розробка та відлагодження програми розв'язання практичної задачі.

7. Методи навчання

Для проведення лекційних занять використовуються наступні методи навчання лекція, розповідь, пояснення, роз'яснення, дискусія.

Для проведення практичних занять використовуються наступні методи навчання інструктаж, ілюстрація, демонстрація, дослід, вправи, навчально-продуктивний пошук, мозковий штурм, мудра порада, коло висновків.

8. Види контролю та методи їх проведення

Поточно-модульний контроль здійснюється два рази за семестр (один раз у рамках вивчення відповідного блоку змістових модулів) та оцінюється за трьома складовими: практичний модульний контроль, теоретичний модульний

контроль і модульний контроль виконання ІЗ (у другому блоці змістових модулів). Поточний контроль з дисципліни здійснюється шляхом вибіркового опитування на початку лекцій, проведення комп'ютерного тестування, експрес-опитування на практичних заняттях.

Оцінка за практичну складову модульного контролю виставляється за результатами оцінювання знань студента під час захисту практичних завдань та ІЗ.

Теоретичний модульний контроль здійснюється у письмовій формі, у вигляді експрес-опитувань і проміжного тестового контролю – модульних тестових контрольних робіт згідно з графіком навчального процесу.

Оцінка за теоретичну складову виставляється за результатами контрольних опитувань і тестового контролю.

Оцінка за ІЗ виставляється за результатами оцінювання знань студента під час захисту індивідуальних завдань і враховує своєчасність їх виконання.

Для підведення підсумків роботи студентів із блоку змістових модулів виставляється підсумкова оцінка, яка складається з оцінок за поточно-модульний контроль.

Модульна контрольна робота проводиться 2 рази за семестр, здійснюється шляхом комп'ютерного тестування та включає тестові завдання різного рівня складності відповідно до тем блоку змістових модулів.

Оцінка за перший блок змістових модулів виставляється студенту за результатами його поточної успішності та тестування за блоком змістових модулів 1.

Другий блок змістових модулів оцінюється за результатами поточної успішності та тестування за блоком змістових модулів 2.

Екзамен (підсумковий контроль) забезпечує оцінку результатів навчання студентів на заключному етапі їх навчання і проводиться відповідно до навчального плану за результатами вивчення дисципліни. Підсумковий контроль проводиться у вигляді екзамену в термін, визначений графіком-календарем навчального процесу та в обсязі навчального матеріалу, визначеному робочою програмою навчальної дисципліни. Кожен екзаменаційний білет складається із 2-х практичних задач, які передбачають вирішення типових професійних завдань фахівця на робочому місці та дозволяють діагностувати рівень теоретичної підготовки студента і рівень його компетентності з навчальної дисципліни. Для розв'язання кожної задачі потрібно розробити програму (проект) на алгоритмічній мові C/C++. Підсумкова оцінка за екзамен є сумою оцінок за кожне завдання.

9. Політика курсу

Для успішного проходження курсу та складання контрольних заходів необхідним є вивчення навчального матеріалу за кожною темою. Специфіка курсу здебільшого передбачає акцент на розумінні підходів і принципів, отримання практичних навичок, а не просто запам'ятовування визначень.

Кожен студент повинен ознайомитися і слідувати Положенню про академічну доброчесність, Статуту і розпорядку дня ХНУПС.

Для успішного засвоєння програмного матеріалу студент зобов'язаний:

- не запізнюватися на заняття;
- не пропускати заняття, а в разі пропуску відновити за допомогою консультування з викладачем та з використанням Moodle конспект, самостійно вивчити матеріал пропущеного заняття та скласти відповідні контрольні заходи в індивідуальному порядку;
- конструктивно підтримувати зворотній зв'язок на всіх заняттях;
- брати активну участь у освітньому процесі;
- своєчасно і старанно виконувати завдання для самостійної роботи;
- не користуватися мобільним телефоном під час занять;
- бути доброзичливим до однокурсників та викладачів;
- брати участь у контрольних заходах;
- будь-яке копіювання або відтворення результатів чужої праці (у тому числі списування), якщо тільки робота не має груповий формат, використання чужих завантажених з Інтернету матеріалів кваліфікується як порушення норм і правил академічної доброчесності та передбачає притягнення винного до відповідальності, у порядку, визначеному чинним законодавством та Положенням про академічну доброчесність ХНУПС.

Результатом невиконання та / або недотримання правил може бути оцінка «незадовільно» за курс.

9.1 Вказівки до самостійної роботи

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною літературою. Самостійна робота є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять. Самостійна робота студентів передбачає поглиблене вивчення тем з використанням рекомендованої літератури, пошук інформації в Інтернеті, а також додаткову роботу в комп'ютерних класах для виконання індивідуальних завдань.

Зміст самостійної роботи:

1. Поглиблене вивчення питань тем за методичними вказівками викладача.
2. Систематичне опрацювання лекційного матеріалу, запропонованої базової та допоміжної літератури з питань курсу.

3. Систематична підготовка до практичних видів занять відповідно до запропонованих планів їх проведення (переліку питань), консультацій викладача, змісту навчальної дисципліни в Moodle.

4. Самостійна підготовка до контрольних заходів.

5. Вивчення основних термінів та понять з галузі обчислювальної техніки і програмування.

6. Робота над ІНДЗ.

У процесі підготовки за темами (питаннями) курсу, що віднесені для самостійного опрацювання, студентам слід звернути особливу увагу на засвоєння зазначених ключових понять та термінів. Обов'язковим вважається ведення студентами конспекту, який повинен містити тезисний огляд питань, що віднесені для самостійного опрацювання, а також визначення ключових понять і термінів.

У процесі вивчення курсу для поточного контролю самостійної роботи курсантів викладач використовує:

- опитування студентів під час практичних видів занять, перевірку робочих конспектів;
- оцінку виконання встановлених нормативів та набуття практичних навичок;
- оцінку вмінь працювати у команді;
- проведення поточних тестувань, проміжного та підсумкового контролю.

9.2 Критерії оцінювання знань

Оцінювання результатів підсумкового контролю навчальних досягнень курсантів здійснюється за 100-бальною шкалою, за шкалою ЄКТС та національною шкалою.

Оцінювання за шкалою ECTS здійснюється згідно табл. 1.

Таблиця 1 - Шкала оцінювання: 100-бальна, ECTS та національна

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
80 - 89	B	добре	
65 - 79	C		
55 - 64	D	задовільно	
50 - 54	E		
35 - 49	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

Критерії оцінювання знань і вмінь студента за результатами вивчення навчального матеріалу модуля (навчальної дисципліни) наступні:

A - оцінка "відмінно" виставляється за високий рівень знань (допускаються деякі неточності) навчального матеріалу модуля, що міститься в основних і додаткових рекомендованих джерелах, вміння аналізувати явища, які вивчаються, у їхньому взаємозв'язку і розвитку, чітко, лаконічно, логічно, послідовно відповідати на поставлені запитання, вміння застосовувати теоретичні положення при розв'язанні практичних завдань;

B - оцінка "дуже добре" виставляється за знання навчального матеріалу модуля вище середнього рівня, включаючи розрахунки, аргументовані відповіді на поставлені запитання (можлива невелика кількість неточностей), вміння застосовувати теоретичні положення під час розв'язання практичних завдань;

C - оцінку "добре" виставляється в цілому за правильне розуміння навчального матеріалу модуля, включаючи розрахунки, аргументовані відповіді на поставлені питання, які, однак, містять певні (несуттєві) недоліки, за вміння застосовувати теоретичні положення при розв'язанні практичних завдань;

D - оцінка "задовільно" виставляється за посередні знання навчального матеріалу модуля, малоаргументовані відповіді, слабе застосовування теоретичних положень під час розв'язання практичних завдань;

E - оцінка "достатньо задовільно" виставляється за слабкі знання навчального матеріалу модуля, неточні або малоаргументовані відповіді, з порушенням послідовності його викладення, за слабе застосовування теоретичних положень під час розв'язання практичних завдань;

FХ - оцінка "незадовільно" з можливістю повторного складання екзамену, виставляється за незнання значної частини навчального матеріалу модуля, суттєві помилки у відповідях на питання, невміння застосовувати теоретичні положення при розв'язанні практичних завдань;

F - оцінка "незадовільно" з обов'язковим повторним вивченням модуля (навчальної дисципліни) виставляється за незнання значної частини навчального матеріалу модуля, суттєві помилки у відповідях на запитання, невміння орієнтуватися при розв'язанні практичних завдань, незнання основних фундаментальних положень.

Таблиця 2 - Розподіл балів за змістовими модулями

Поточне тестування та самостійна робота												Екзамен	Сума
Блок змістових модулів 1						Блок змістових модулів 2							
ЗМ 1.1	ЗМ 1.2	ЗМ 1.3	ЗМ 1.4	ЗМ 1.5	ЗМ 1.6	ЗМ 2.1	ЗМ 2.2	ЗМ 2.3	ЗМ 2.4	ЗМ 2.5	ЗМ 2.6		
5	5	5	5	5	5	5	5	5	5	5	5	40	100

В основу системи оцінювання покладено поопераційний контроль і накопичення рейтингових балів за різнобічну навчально-пізнавальну діяльність студентів у процесі навчання.

Метою оцінювання є:

- інтенсифікація навчального процесу та підвищення якості підготовки фахівців;
- підвищення мотивації студентів до активного, свідомого навчання, систематичної самостійної роботи протягом семестру та відповідальності за результати навчальної діяльності;
- встановлення постійного зворотного зв'язку з кожним студентом та своєчасне коригування його навчальної діяльності;
- забезпечення змагальності та здорової конкуренції у навчанні;
- підвищення об'єктивності оцінювання рівня підготовки студентів;
- зменшення психологічних, емоційних і фізичних перевантажень у період екзаменаційних сесій.

10. Методичне забезпечення та рекомендована література

Основна

1. Вирт Н. Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. – Москва : Мир, 1989. – 360 с. : ил.
2. Кнут Д. Искусство программирования для ЭВМ / Д. Кнут. – Москва : Мир,
Т. 1 – 1997
Т. 2 – 1997
Т. 3 – 1998
3. Страуструп Б. Программирование: принципы и практика использования C++ / Б. Страуструп; пер. с англ. – 2-е изд. – Москва : Издательский дом “Вильямс”, 2016. – 1328 с.
4. Деннис М. Ритчи. Язык программирования C / Деннис М. Ритчи, Брайан У. Керниган ; пер. с англ. – Москва : Издательский дом “Вильямс”, 2009. – 304 с.
5. Ишкова Э. А. C++. Начала программирования / Э. А. Ишкова. – Москва : ООО “БИНОМ-Пресс”, 2004. – 368 с. : ил.
6. Хортон А. Visual C++ 2010: полный курс / А. Хортон ; пер. с англ. – Москва : Издательский дом “Вильямс”, 2011. – 1216 с. : ил.
7. Дейтел Х. Как программировать на C++ / Х. Дейтел, П. Дейтел ; пер. с англ. – Москва : ЗАО “Издательство БИНОМ”, 2008. – 1455 с.
8. Аммерааль Л. STL для программистов на C++ / Л. Аммерааль ; пер. с англ. – Москва : ДМК, 1999. – 240 с. : ил.
9. Щупак Ю. А. Эффективная разработка приложений / Ю. А. Щупак. – Санкт-Петербург : Питер, 2007. – 572 с. : ил.
10. Румянцев П. В. Азбука программирования в Win32 API / П. В. Румянцев. – Москва : Горячая линия – Телеком, 2001. – 312 с.
11. Пирогов В. Ю. Программирование на Visual C++.NET / В. Ю. Пирогов. – Санкт-Петербург : БХВ-Петербург, 2003. – 800 с.

Додаткова

12. Страуструп Б. Язык программирования C++. Специальное издание / Б. Страуструп ; пер. с англ. – 3-е изд. – Санкт-Петербург; Москва : “Невский диалект” ; Изд-во “БИНОМ”, 2004. – 1104 с. : ил.
13. Подбельский В. В. Язык C++ : учеб. пособ. / В. В. Подбельский. – 4-е изд. – Москва : Финансы и статистика, 1999. – 560 с.
14. Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – Санкт-Петербург : Символ-Плюс, 2009. – 608 с. : ил.
15. Павловская Т. А. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. – Санкт-Петербург : Питер, 2006. – 461 с. : ил.
16. Саймон Р. Windows 2000 API. Энциклопедия программиста / Р. Саймон ; пер. с англ. – Киев : ООО “ДиасофтЮП”, 2002. – 1088 с.

17. Марченко А. Л. С++. Бархатный путь / А. Л. Марченко. – Москва : Горячая линия – Телеком, 1999. – 400 с.
18. Подбельский В. В. Программирование на языке С / В. В. Подбельский, С. С. Фомин. – 2-е доп. изд. – Москва : Финансы и статистика, 2004. – 600 с.
19. Шеферд Д. Программирование на Microsoft Visual C++.NET / Д. Шеферд ; пер. с англ. – Москва : Издательско-торговый дом “Русская редакция”, 2003. – 928 с. : ил.

11. Інформаційні ресурси

20. Главная страница MSDN, MSDN по-русски [Электронный ресурс]. – Режим доступа : <http://msdn.microsoft.com/ru-ru/default.aspx>.
21. Интернет Университет Информационных Технологий – дистанционное образование [Электронный ресурс]. – Режим доступа : <http://www.intuit.ru/>.
22. CodeNet – все для программиста [Электронный ресурс]. – Режим доступа : <http://www.codenet.ru/>.
23. REALCODING – для настоящих программистов [Электронный ресурс]. – Режим доступа : <http://www.realcoding.net/>.
24. MSDN-WindowsAPI. [Электронный ресурс]. – Режим доступа : http://www.vsokovikov.narod.ru/New_MSDN_API/index_msdn.htm.