

26
26.6

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОПЕТРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ОЛЕСЯ ГОНЧАРА
Факультет прикладної математики
Кафедра математичного забезпечення ЕОМ

«Затверджую»
Ректор Дніпропетровського
національного університету
ім. О. Гончара М.В. Поляков
«__» _____ 201_ р.



«Погоджено»
Проректор
з науково-педагогічної роботи
С.О. Чернецький
«__» _____ 201_ р.

ПРОГРАМА
ФАХОВОГО ВСТУПНОГО ВИПРОБУВАННЯ
за рівнем бакалавра
121 «Інженерія програмного забезпечення»
прискорена форма навчання

Розглянуто на Вченій раді ФПМ
протокол № 3 від 29.11.2016 р.

Голова Вченої ради ФПМ
проф. О.М. Кісельова
«29» _____ 2016 р.

Завідувач кафедри
математичного забезпечення ЕОМ
О.Г. Байбуз
«28» _____ 11 _____ 2016 р.

Дніпро
2016

ПЕРЕЛІК ДИСЦИПЛІН, ЗА ЯКИМИ БУДУТЬ АТЕСТУВАТИСЬ АБИТУРІЄНТИ:
Основи програмування та алгоритмічні мови, Операційні системи, Архітектура комп'ютерів, Основи програмної інженерії, Об'єктно-орієнтоване програмування.

В результаті вивчення дисциплін фахівець повинен знати:
основи алгоритмізації та програмування, вимоги до програмної системи та потреби користувача, види та методи абстракції в програмуванні, сучасні уявлення про структуру та архітектуру програмного забезпечення в програмній інженерії, базові уявлення про об'єктно-орієнтований підхід у розробці програмного забезпечення, методи та технології об'єктно-орієнтованого програмування, можливості апаратного забезпечення, можливості операційних систем.

В результаті вивчення дисциплін фахівець повинен вміти:

- визначати джерела вимог і забезпечувати процес їх витягання;
- розробляти специфікації вимог користувачів;
- здійснювати аналіз вимог, розробляти специфікацію програмних вимог, виконувати їхню верифікацію та атестацію;
- моделювати різні аспекти системи, для якої створюються ПЗ;
- проектувати компоненти архітектурного рішення;
- володіти основами конструювання ПЗ;
- володіти методами та технологіями організації та застосування даних;
- володіти методами та технологіями об'єктно-орієнтованого програмування;
- використовувати можливості апаратного забезпечення;
- використовувати можливості операційних систем.

Структура тестового завдання

Тестове завдання включає тести наступної форми:

завдання з вибором однієї правильної відповіді з чотирьох можливих – 2 бали; 0 балів, якщо вказано неправильну відповідь, або вказано більше однієї відповіді, або відповіді не надано.

Кожен білет складається з 50 тестів за наступними пропорціями:

15 тестів	з дисципліни	Основи програмування та алгоритмічні мови (Блок 1)
7 тестів	з дисципліни	Операційні системи (Блок 2)
6 тестів	з дисципліни	Архітектура комп'ютера (Блок 3)
7 тестів	з дисципліни	Основи програмної інженерії (Блок 4)
15 тестів	з дисципліни	Об'єктно-орієнтоване програмування (Блок 5)

Максимальна можлива загальна сума дорівнює 100 балам та складається як $50 \cdot 2 = 100$.

ОСНОВИ ПРОГРАМУВАННЯ ТА АЛГОРИТМІЧНІ МОВИ

1. Алгоритми роботи програм, їх типи, способи опису алгоритмів, їх властивості.
2. Абетка мови C, перелік операцій, структура програми.
3. Зарезервовані ідентифікатори та ключові слова.
4. Константи, їх типи, правила використання коментарів.
5. Вирази, їх типи та правила конструювання.
6. Типи змінних, опис змінних, класи пам'яті.
7. Типи розгалужень. Синтаксис та правила використання операторів if, if-else.
8. Синтаксис та правила використання оператора вибору switch.
9. Типи циклів. Оператор for.
10. Принципи та правила обробки масивів, їх ініціалізація.
11. Синтаксис та правила використання операторів do-while, while.
12. Синтаксис та правила використання continue та break.
13. Вказівники, ініціалізація вказівників, динамічна пам'ять.
14. Динамічні матриці, правила їх опису та використання.
15. Рядки, масиви символів. Представлення рядків у пам'яті, функції обробки рядків.
16. Параметри функцій, параметри-масиви. Оператор return.
17. Структури, правила ініціалізації структур та доступу до елементів структур.
18. Файли, стандартні файли. Відкриття файлу. Читання/запис із/у файл.
19. Довільний доступ до файлів. Запис структур у файли.

ОПЕРАЦІЙНІ СИСТЕМИ

1. Поняття операційної системи (ОС), її призначення та функції.
2. Архітектурні особливості операційних систем.
3. Функціональні компоненти операційних систем.
4. Базові поняття процесів і потоків.
5. Загальні принципи керування пам'яттю.
6. Керування файлами і пристроями.
7. Цілі і завдання файлової системи.
8. Логічна організація файлової системи.
9. Фізична організація файлової системи.
10. Засоби організації файлових операцій.
11. Командний рядок Windows. Консольні команди Windows.
12. Огляд мережних функцій ОС.

АРХІТЕКТУРА КОМП'ЮТЕРІВ

1. Способи організації та типи обчислювальних систем.
2. Представлення даних.
3. Поняття архітектури і структури комп'ютера.
4. Процесори.
5. Організація і функціонування систем пам'яті.
6. Система введення-виведення обчислювальної системи та її структура.
7. Структура програм на мові асемблера.
8. Способи адресації.
9. Регістри мікропроцесора.
10. Функціональна класифікація машинних команд та директиви.
11. Організація введення та виведення даних на мові асемблера.
12. Арифметичні, логічні операції, побітові операції та операції зсуву на мові асемблера.

ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Програмна інженерія та її місце серед інженерних дисциплін.
2. Етапи процесу розробки програмного забезпечення. Поняття життєвого циклу програмного забезпечення.
3. Класична (каскадна) модель процесу розробки програмного забезпечення.
4. Макетування при розробці програмного забезпечення. Інкрементна модель процесу розробки програмного забезпечення.
5. Поняття ризику процесу розробки програмного забезпечення. Еволюційна (спіральна) модель процесу розробки програмного забезпечення.
6. Сутність об'єктно-орієнтованого підходу до розробки програмного забезпечення.
7. Життєвий цикл програмного забезпечення при об'єктно-орієнтованому підході.
8. Аналіз вимог до програмного забезпечення: види вимог.
9. Аналіз вимог до програмного забезпечення: діаграми варіантів використання (моделі сценаріїв) програмного забезпечення.
10. Поняття специфікації програмних вимог з використанням UML.
11. Документування моделей предметної області з використанням UML. Статистичні моделі предметної області об'єктно-орієнтованих програмних систем – діаграма класів.
12. Динамічні моделі об'єктно-орієнтованих програмних систем: діаграма станів та діаграма діяльності.
13. Динамічні моделі об'єктно-орієнтованих програмних систем: діаграма взаємодії та діаграма послідовності.
14. Технології повторного використання як фундамент програмної інженерії. Повторне використання програмних компонентів.
15. Введення в тестування програмного забезпечення. Тестування на етапі проектування програмного забезпечення.

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

1. Проект у MVC++.
2. Поняття класу. Члени класу.
3. Структура та опис класу.
4. Інкапсуляція та приховування інформації. Специфікатори доступу.
5. Поняття об'єкту. Оператори доступу до членів об'єкту.
6. Конструктори та деструктори.
7. Вказівник this.
8. Поліморфізм.
9. Перевантаження функцій.
10. Дружні функції. Дружні класи.
11. Перевантаження операторів.
12. Успадкування та повторне використання коду.
13. Множинне успадкування. Перевизначення методу базового класу.
14. Віртуальні функції. Механізм пізнього зв'язування.
15. Абстрактні класи.
16. Статистичні елементи класу.
17. Відношення між класами
18. Шаблони функцій.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ:

1. Павловська Т.А. Программирование на языке высокого уровня C/C++: учеб. – М. – СПб.: Питер, 2003. – 460 с.
2. Павловська Т.А. Программирование на языке высокого уровня C/C++. Практикум: учеб. / Т.А. Павловська. – М. – СПб.: Питер, 2003. – 238 с.
3. Прата С. Язык программирования С. Лекции и упражнения. – М.: Diasoft, 2002. – 883 с.
4. Ковалюк Т.В. Основи програмування: підручник для вищих навчальних закладів / Т.В.Ковалюк. – К.: Видавнича група ВНУ, 2005. – 384 с.
5. Довбуш Галина. Visual C++ на примерах / Галина Довбуш, Анатолий Хомоненко. – СПб.: БХВ-Петербург, 2008. – 511 с.
6. Шеховцов В.А. Операционные системы. – К.: Видавнича група ВНУ, 2005. – 576 с.
7. Уильям Р. Microsoft Windows XP Professional: справочник администратора – пер. с англ. – 2-е изд. – М.: Издательско-торговый дом «Русская редакция», 2003. – 448 с.
8. Таненбаум, Эндрю. Современные операционные системы. – 3-е изд. – М. – СПб.: Питер, 2011. – 1120 с.
9. Таненбаум, Эндрю. Архитектура компьютера. – М. СПб.: Питер, 2016. – 816 с.
10. Юров В.И. Assembler. – СПб.: Питер, 2010. – 640 с.
11. Гуров В.В. Основы теории и организации ЭВМ: [учеб. пособие] / Валерий Гуров, Всеволод Чуканов. – М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2006. – 272 с.
12. Введение в программную инженерию и управление жизненным циклом программного обеспечения – Guide to Software Engineering Base of Knowledge (SWEBOOK): Пер. с англ. С.Орлик. – М., 2005.
13. Соммервилл Иан. Инженерия программного обеспечения: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 624 с.: ил.
14. Леоненков А. Самоучитель UML. – СПб.: БХВ-Петербург, 2007. – 576 с.
15. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 384 с.: ил.
16. Мейер Б., Бодуэн К. Методы программирования. – М.: Мир, 1982, Т. 1. – 356 с.
17. Мейер Б., Бодуэн К. Методы программирования. – М.: Мир, 1982, Т. 2. – 368 с.
18. Бадд Тимоти. Объектно-ориентированное программирование в действии. – СПб.: Питер, 1997. – 446 с.
19. Страуструп Б. Язык программирования C++. Специальное издание. – СПб. – М.: «Невский диалект» – «БИНОМ», 2006. – 1104 с.