

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ОЛЕСЯ ГОНЧАРА

ЗАТВЕРДЖУЮ

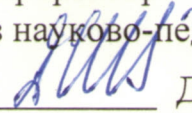
В.О. ректора


Ольга СОКОЛЕНКО

« 25 » _____ 2021 р.

ПОГОДЖЕНО

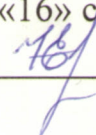
Проректор
з науково-педагогічної роботи

 Дмитро СВИНАРЕНКО

« ____ » _____ 2021 р.

ПРОГРАМА
ФАХОВОГО ВСТУПНОГО ВИПРОБУВАННЯ
для вступу на навчання за освітнім рівнем бакалавра
на основі освітньо-кваліфікаційного рівня молодшого спеціаліста
121 «Інженерія програмного забезпечення»
прискорена форма навчання

Розглянуто на засіданні вченої ради
факультету прикладної математики
«16» січня 2021 р., протокол № 5

Голова вченої ради  (Олена КІСЕЛЬОВА)

Дніпро
2021

Укладачі програми:

1. Байбуз Олег Григорович, завідувач кафедри математичного забезпечення ЕОМ.
2. Антоненко Світлана Валентинівна, доцент кафедри математичного забезпечення ЕОМ.

Програма ухвалена на засіданні кафедри математичного забезпечення ЕОМ
«11» січня 2021 р., протокол № 7

Завідувач кафедри  (Олег БАЙБУЗ)
(підпис)

ПЕРЕЛІК ДИСЦИПЛІН, ЗА ЯКИМИ БУДУТЬ АТЕСТУВАТИСЬ АБІТУРІЄНТИ:
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. Mark Reed C++: The Ultimate Beginners Guide to Learn C++ Programming. – Kindle Edition, 2021. – 258 p.
2. Ковалюк Т.В. Основи програмування: підручник для вищих навчальних закладів / Т.В.Ковалюк. – К.: Видавнича група BHV, 2005. – 384 с.
3. Довбуш Г. Visual C++ у прикладах / Галина Довбуш, Анатолій Хомоненко. – К.: Видавнича група BHV, 2008. – 511 с.
4. Шеховцов В.А. Операційні системи. – К.: Видавнича група BHV, 2005. – 576 с.
5. Stanek, William R. Microsoft Windows XP Professional, Engl. ed. – Microsoft Press Books, 2001 – 353 p.
6. Andrew S. Tanenbaum. Distributed Operating Systems 1st Edition – Pearson, 1994. – 632 p.
7. Andrew S. Tanenbaum. Structure Computer Organisation 5st Edition. – Pearson, 2005. – 800 p.
8. Scanlon L.J. IBM PC & XT Assembly Language. – New York: Brady, 1985. – 416 p.
9. John L. Hennessy Computer Organization and Design MIPS Edition: The Hardware/Software Interface. – Morgan Kaufmann, 2013. – 800 p.
10. IEEE Computer Society. Guide to the Software Engineering Body of Knowledge // Pierre Bourque, Richard E. Fairley. – IEEE Computer Society Press, 2014. – 346 p.
11. Ian Sommerville. Software Engineering. – Pearson, 2015. – 816 p.
12. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language 3rd Edition. – Addison-Wesley Professional, 2003. – 208 p.
13. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы: Пер. с англ. – К.: Видавнича група «Діалектика», 2001. – 384 с.: ил.
14. Мейер Б., Бодуэн К. Методы программирования. – М.: Мир, 1982, Т. 1. – 356 с.
15. Мейер Б., Бодуэн К. Методы программирования. – М.: Мир, 1982, Т. 2. – 368 с.
16. Timothy Budd. Introduction to Object-Oriented Programming. – Pearson, 2001. – 640 p.
17. Bjarne Stroustrup. The C++ Programming Language: Special Edition. – Addison-Wesley Professional, 2000. – 1030 p.