

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М.Є. Жуковського  
«Харківський авіаційний інститут»

**ЗАТВЕРДЖЕНО**

вченою радою  
Національного аерокосмічного  
університету ім. М.Є. Жуковського  
«Харківський авіаційний інститут»

Заступник голови вченої ради  
О.В. Гайдачук

21 лютого 2018 р. протокол № 7



**ПРОГРАМА  
ДОДАТКОВОГО ВСТУПНОГО ВИПРОБУВАННЯ**

для здобуття освітнього ступеня магістра  
за освітньо-професійною програмою  
зі спеціальності

**113 Прикладна математика**

(код та найменування)

(освітні програми Математичне та комп'ютерне моделювання,

Обчислювальний інтелект)

(найменування)

**у 2018 році**

Харків  
2018

## ВСТУП

Додаткове вступне випробування для здобуття освітнього ступеня магістра за освітньо-професійною програмою зі спеціальності

113 Прикладна математика

(код та найменування)

(освітні програми Математичне та комп'ютерне моделювання, Обчислювальний інтелект)

(найменування)

відбувається відповідно до «Правил прийому до Національного аерокосмічного університету ім. М.Є. Жуковського «Харківський авіаційний інститут» у 2018 році» у формі індивідуального письмового фахового іспиту, який приймає екзаменаційна комісія з певної спеціальності (освітньої програми), склад якої затверджується наказом ректора Університету.

До додаткового фахового іспиту входять питання за темами:

- дискретна математика,
- об'єктно-орієнтоване програмування.

Перелік питань за темами наведений у програмі.

### **Критерії оцінювання знань**

1. Результат додаткового фахового іспиту визначається за 100-бальною шкалою. При отриманні вступником 60 балів та більше він допускається до вступного випробування.

2. Екзаменаційний білет складається з 20-ти закритих тестових завдань (по 10 завдань з тем «Дискретна математика» та «Об'єктно-орієнтоване програмування»). Серед запропонованих у білеті відповідей на тестове завдання вступнику слід обрати одну або декілька правильних. Для завдань, в яких потрібно вибрати декілька варіантів відповідей, бали будуть зараховані лише в тому випадку, якщо всі відповіді на запитання, вибрані абітурієнтом, будуть вірними. Правильна відповідь на тестове завдання оцінюється у 5 балів, а неправильна – у 0 балів.

# 1 Питання за темою Дискретна математика

(найменування)

## 1.1 Множини, відношення, функції.

Предмет курсу. Основні поняття дискретної математики; її місце в загальному курсі математики. Елементи і множини. Способи завдання множин в ЕОМ в вигляді булевих масивів. Дискретні множини. Об'єднання, перетин, доповнення множин. Використання включення-виключення для рішення теоретико-числових задач. Завдання множин списками. Операції над множинами. Завдання множин в ЕОМ у вигляді масиву величин булевого типу. Перелік елементів множин. Перелік підмножин з допомогою алгоритму Грея. Відношення порядку. Доповнення часткового порядку на множині до лінійного. Уявлення множин списками.

## 1.2 Алгоритми сортування одновимірних масивів.

Класифікація алгоритмів сортування. Алгоритми злиття. Алгоритм Heapsort. Аналіз, структура програми. Швидке (бінарне) сортування. Аналіз, застосування. Способи сортування за лінійний час. Реалізація на ЕОМ алгоритмів впорядкування множин злиттям списків множин. Сортування бази даних швидкими методами. Швидке (бінарне) сортування. Сортування за лінійний час.

## 1.3 Елементи комбінаторики.

Група підстановок. Транспозиції, інверсії; парність, непарність. Спеціальні засоби генерування підстановок. Перестановки. Реалізація на ЕОМ. Розміщення, сполуки. Розв'язання найпростіших оптимізаційних задач. Біноміальні коефіцієнти. Їхні властивості; дослідження з застосуванням методів математичного аналізу. Числа Стірлінга, Белла. Твірні функції. Біноміальні коефіцієнти. Їхні властивості; дослідження з використанням методів математичного аналізу. Біном Ньютон. Пошук твірних функцій. Застосування твірних функцій для розв'язання комбінаторних задач. Методи програмування для генерування комбінаторних конфігурацій. Генерування розміщень. Задача комівояжера. Генерування сполук, сполук з повтореннями

## 1.4 Алгебраїчні структури.

Операції і алгебри. Класифікація властивостей операцій. Морфізми. Алгебри з однією операцією. Полугрупи, моноїди, групи. Пошук циклів в підстановках. Побудова фактор-груп за циклічними підгрупами. Цикли. Програмна реалізація пошуку циклів. Елементи „довгої арифметики”. Дії «довгої арифметики».

## 1.5 Дерева.

Дерева, як вид графів. Основні властивості дерев. Орієнтовані, впорядковані і бінарні дерева. Дерева. Основні властивості дерев. Орієнтовані, упорядковані і бінарні дерева. Уявлення графів в ЕОМ. Уявлення вільних, орієнтованих і упорядкованих дерев. Обхід бінарних дерев. Уявлення графів в ЕОМ. Уявлення вільних, орієнтованих і упорядкованих дерев. Програмування обходу бінарних дерев. Дерева сортування. Записи, ключі. Способи реалізації асоціативної пам'яті. Алгоритм бінарного пошуку. Алгоритм пошуку в дереві сортування. Алгоритми вилучення і вставки в

дерево сортування. Алгоритм симетричного обходу бінарного дерева з використанням стеку. Вирівненні дерева. Збалансовані дерева.

#### 1.6 Кодування і криптографія

Загальна постановка задачі кодування і декодування. Алфавитне кодування. Алгоритми вилучення і вставки в дерево сортування. Префікс і постфікс слова. Префіксні схеми. Вирівняні дерева. Збалансовані дерева. Кодування з мінімальною надлишковістю. Мінімізація довжини коду повідомлення. Ціна кодування. Алгоритм Фано. Найпростіші методи кодування. Алфавитне кодування. Оптимальне кодування. Побудова бінарного дерева декодування. Ціна кодування. Алгоритм Фано. Програмна реалізація алгоритму Фано і рекурсивного алгоритму Хафмена. Ціна кодування. Алгоритм Фано. Перешкодостійке кодування. Класифікація помилок. Кодова відстань. Код Хемінга. Програмна реалізація алгоритму Фано. Стиск даних. Стиск тексту. Побудова словника. Адаптивний стиск. Перешкодостійке кодування. Код Хемінга; програмування. Структура дерева для побудови словника. Приклад. Шифрування з допомогою випадкових чисел. Постановка задачі шифрування. Шифрування з допомогою випадкових чисел. Криптостійкість. Програмна реалізація шифрування Модулярна арифметика. Теорема Ейлера; мала теорема Ферма. Теорема про залишки. Модулярна арифметика. Теорема про залишки. Шифрування з відкритим ключем. Криптосистема RSA (Рівест, Шамір, Адлеман). Цифровий підпис.

#### Література

1. Применение ЭВМ для решения задач дискретной математики / Соколов А.Ю., Угрюмов М.Л., Халтурин В.А., Чернышев Ю.К.; Слепичева М.А. Харьков : «ХАИ», 2003. – 78 с.
2. Чернышев Ю.К. Вычислительные задачи дискретной математики/ Слепичева М.А.; –Харьков : «ХАИ», 2003. – 72 с.
3. Новиков Ф.А. Дискретная математика для программистов / Новиков Ф.А.; –СПб.: Питер, 2000. – 304 с.
4. Липский В. Комбинаторика для программистов / Липский В.; М.: Мир, 1988.–212с.
5. Чернышев Ю.К., Методы программной реализации событийного моделирования в задачах физики и социометрии. Учебное пособие / Ю.К. Чернышев, О. В. Хайленко, А.В. Товстик – Х.:нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2015. – 64с.

Питання склав  
к.т.н, професор  
(науковий ступень, посада)



Ю.К. Чернишов  
(ініціали та прізвище)

## 2 Питання за темою Об'єктно-орієнтоване програмування

(найменування)

### 2.1 Основні положення об'єктно-орієнтованого підходу

Платформа .NET. .NET Framework. Компіляція і мова MSIL. Об'єктно-орієнтована мова програмування C#. Класи, об'єкти і типи. Розробка проекту консольної програми. Використання від лагодження у середовищі Microsoft Visual Studio.NET. Основи мови програмування C# (типи, змінні та константи, вирази, пробільні символи, оператори, операції, простори імен, директиви препроцесора). Прості та складні програмні системи. Декомпозиція програмних систем. Способи декомпозиції. Об'єктно-орієнтована декомпозиція. Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом. Об'єктно-орієнтований аналіз та його мета. Поняття предметної області. Головні види вимог до програмної системи. Об'єктно-орієнтоване проектування. Елементи архітектури додатку. Визначення класів предметної області. Принципи проектування класів. Об'єктно-орієнтоване програмування. Принципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм.

### 2.2 Класи і об'єкти

Визначення класів. Створення об'єктів. Статичні члени класу. Знищення об'єктів. Передача параметрів. Перевантаження методів і конструкторів. Інкапсуляція даних у властивостях. Поля, що призначені тільки для читання.

### 2.3 Масиви, індексатори і класи колекцій

Масиви. Оператор foreach. Індексатори. Інтерфейси класів колекцій. Клас ArrayList. Черги. Стек. Словники.

### 2.4 Основи об'єктно-орієнтованого проектування мовою UML

Моделювання програмного забезпечення. Методи об'єктно-орієнтованого аналізу і проектування. Мова UML. UML-діаграми. Діаграми класів. Генерування програмного коду. Платформа Microsoft .NET: історія розвитку, архітектура, засоби розроблення додатків, компіляція та виконання програм, бібліотека базових класів, система типізації. Загальні відомості про мову C#: особливості використання, алфавіт, типи даних, порівняння типів-значень та типів-посилань, вбудовані типи-значення, вбудовані типи-посилання, одновимірні та багатовимірні масиви, операції, оператори, структура програми, коментарі, особливості використання функцій, механізми передачі параметрів, простори імен, основи використання бібліотеки базових класів .NET Культура програмування: вимоги до інтерфейсу користувача та вихідного тексту програми.

### 2.5 Наслідування і поліморфізм

Спеціалізація та узагальнення. Наслідування. Поліморфізм. Абстрактні класи. Клас Object – коріння всіх класів. Упаковка і розпаковка типів. Вкладені класи. Абстрактні типи даних. Проектування Абстрактного типу даних. Синтаксис структур та класів у мові C#. Елементи класу. Доступ до елементів класу. Посилання this. Перевантаження методів класу. Об'єкти в програмі. Послідовність створення об'єкта. Конструктори. Основні властивості конструкторів. Звільнення пам'яті. Система "збору сміття". Статичні дані та методи: призначення, властивості, особливості використання. Відношення

агрегації. Реалізація агрегації у мові C#. Відношення спадкування. Синтаксис спадкування у мові C#. Ініціалізація об'єкта базового класу. Варіанти використання спадкування. Перевизначення методів. Заборона спадкування. Рядкове представлення об'єкта. Реалізація принципу поліморфізму у мові C#. Раннє та пізнє зв'язування. Віртуальні методи. Абстрактні класи та методи. Реалізація поліморфної поведінки на базі абстрактного класу. Правила застосування абстрактних класів. Інтерфейси. Реалізація поліморфної поведінки на базі інтерфейсу. Правила застосування інтерфейсів. Принципи перевантаження операцій. Особливості використання функції operator. Індексатори. Властивості.

## 2.6 Стандартні бібліотеки класів Microsoft .NET Framework

Джерела та споживачі даних. Загальні відомості про потоки введення-виведення даних. Алгоритми роботи потоків введення-виведення даних. Основні класи стандартної бібліотеки .NET для підтримки введення-виведення даних. Загальні відомості про контейнери. Основні елементи та структури даних стандартної бібліотеки контейнерів .NET Типізовані контейнери. Особливості реалізації рядкового типу даних у платформі .Net Класи стандартної бібліотеки .NET для подання рядків та особливості їхнього використання. Форматування рядків. Призначення та застосування регулярних виразів. Підтримка регулярних виразів у стандартній бібліотеці .NET Спеціальні символи, які використовуються у регулярних виразах. Введення до атрибутів. Елементи програми до яких можливо застосування атрибутів. Визначені атрибути. Використання атрибутів умовної компіляції. Атрибути рівня модулю компіляції. Збереження та відновлення стану об'єктів у .NET Серіалізація та десеріалізація. "Граф" об'єктів при серіалізації. Створення класів, об'єкти яких можливо серіалізувати. Процеси серіалізації та десеріалізації. Формати серіалізації. Серіалізація та десеріалізація об'єктів у двійковому та XML-форматах. Використання вбудованих запитів LINQ.

## 2.7 Розроблення DLL-бібліотек

Бібліотеки та їх використання. Статичні та динамічні бібліотеки. DLL-бібліотеки. Розроблення DLL-бібліотеки на платформі Microsoft .NET.

## 2.8 Структури. Оператори. Інтерфейси

Визначення структур. Загальні риси та відмінності структур від класів. Створення об'єктів структур. Ключове слово – operator. Підтримка інших мов платформи .NET. Створення нових операцій. Логічні пари. Операція перевірки на рівність. Операції приведення типів. Визначення і реалізація інтерфейсу. Звернення до методів інтерфейсу. Перевизначення реалізації інтерфейсу. Явна реалізація інтерфейсу.

## 2.9 Основи використання технології Windows Forms

"Традиційна" модель програмування на платформі .NET Модель "Windows-програмування" на платформі .NET Технологія Windows Forms. Форми. Загальна структура додатку з графічним інтерфейсом користувача на платформі .NET Розроблення додатків Windows Forms за допомогою інтегрованого середовища. Події рівня форми. "Колекція" візуальних елементів управління форми. Використання базових візуальних елементів управління.

2.10 Розроблення графічних інтерфейсів користувача за допомогою технології Windows Forms

Основи архітектури додатків Windows Forms. Модель подій у Windows Forms. Діалогові вікна. Основні візуальні елементи управління: властивості та використання. Компоненти форми для виключення помилкового введення даних користувачем. Використання візуальних елементів управління "дерево" та "таблиця". Робота з даними у XML-форматі.

2.11 Обробка виняткових ситуацій

Види помилок у програмах. Проблеми традиційного підходу до обробки помилок. Механізм обробки виключень. Класи виключень стандартної бібліотеки .NET. Синтаксис обробки виключень. Перевірка на арифметичне переповнення. Об'єкти Exception. Виклик програмованих виключень. Повторний виклик виключення.

2.12 Делегати і події

Загальні відомості про делегати. Оголошення та використання делегатів у мові C#. Анонімні методи. Групові делегати. Загальні відомості про події. Генерування подій.

2.13 Використання графічних можливостей платформи Microsoft .NET

Особливості графічного виведення даних. Логічна система координат. Простори імен GDI+. Обробка повідомлення перемальовування. Програмне генерування повідомлення перемальовування. Графічні об'єкти GDI+. Використання пензлів, пер та шрифтів. Робота з графічними зображеннями.

#### Література

1. Скоб Ю.О. Сучасні технології програмування: Навч. посібник до лаб. практикуму. – Харків: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2011.–100с. (абонемент бібліотеки гол.корп.)

2. Шевчук А. Design Patterns via C#. Приемы объектно-ориентированного проектирования / А.Шевчук, Д. Охрименко, А. Касьянов. – 2015. – 200 с.

3. Троелсен Эндрю Язык программирования C# 6.0 и платформа .NET 4.6 – Изд.: «Вильямс», 2016– 1440с.

4. Коробчинський К. П. ООП: Навчальний курс [Електронний ресурс] / К. П. Коробчинський / Портал ХАІ. – Режим доступу: <http://stm.khai.edu/>

5. Либерти Д. Программирование на C#: Пер. с англ. – СПб: Символ-Плюс, 2003. – 688 с.

Питання склав

СТ. ВИКЛ.

(науковий ступень, посада)



К.П. Коробчинський

(ініціали та прізвище)

Завідувач кафедри 304

(підпис)

А. Г. Чухрай

(ініціали та прізвище)

Програму розглянуто й узгоджено на випусковій кафедрі 304  
Протокол № 7 від «02» січня 2018 р.

Програму додаткового вступного випробування для здобуття освітнього ступеня магістра за освітньо-професійною програмою зі спеціальності 113 Прикладна математика

(код та найменування)

(освітні програми Математичне та комп'ютерне моделювання, Обчислювальний інтелект)

(найменування)

узгоджено науково-методичною комісією Національного аерокосмічного університету ім. М.Є. Жуковського «Харківський авіаційний інститут» з галузей знань «Математика та статистика», «Інформаційні технології», «Автоматизація та приладобудування», «Хімічна та біоінженерія», «Електроніка та телекомунікації»

Протокол № 1 від «08» лютого 2018 р.

Голова НМК 2  
к.т.н., доц.



О.В. Заболотний