

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики

**УТВЕРЖДАЮ**  
декан факультета вычислительной  
математики и кибернетики

**И.А. Соколов /**  
**«27» сентября 2023г.**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине

**Программирование на языке C#**

**Уровень высшего образования:**

**бакалавриат**

**Направление подготовки / специальность:**

**02.03.02 "Фундаментальная информатика и информационные технологии" (3++)**

**Направленность (профиль) ОПОП:**

**Искусственный интеллект и анализ данных**

**Форма обучения:**

**очная**

Рассмотрен и утвержден

*на заседании Ученого совета факультета ВМК*

*(протокол №7, от 27 сентября 2023 года)*

Москва 2023

## 1. ФОРМЫ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

В процессе и по завершении изучения дисциплины оценивается формирование у студентов следующих компетенций:

Планируемые результаты обучения по дисциплине (модулю)		
Содержание и код компетенции.	Индикатор (показатель) достижения компетенции	Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций
ОПК-4. Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла	ОПК-4.1. Участвует в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил ОПК-4.2. управляет проектами создания информационных систем на стадиях жизненного цикла	знать основы программирования для решения задач, базовый синтаксис и общая структура языка программирования C#. принципы построения приложений на языке C# с графическим пользовательским интерфейсом Уметь работать со средой визуального программирования MS Visual Studio, и языком программирования высокого уровня C#. создавать простые, но хорошо документированные и структурированные приложения на языке C# с графическим пользовательским интерфейсом, Владеть разработкой консольных приложений на языке C# для реализации алгоритмов обработки данных разработкой приложений с графическим интерфейсом на языке C#

### 1.1. Текущий контроль успеваемости

Текущий контроль успеваемости осуществляется путем оценки результатов выполнения заданий практических (семинарских) занятий, самостоятельной работы, предусмотренных учебным планом и посещения занятий/активность на занятиях.

В качестве оценочных средств текущего контроля успеваемости предусмотрены:

#### Примерные тестовые задания:

1. Что понимается под простым типом?

- A) тип данных, значения которого не содержат компонент, например, целое или символ, в отличии от массива или структуры;
  - B) тип данных, значения которого содержат компоненты, например массивы, структуры;
  - C) тип данных, значения которого являются массивами;
  - D) тип данных, значения которого являются целыми числами;
2. Что понимается под простым типом?
- A) тип данных, значения которого не содержат компонент, например, целое или символ, в отличии от массива или структуры;
  - B) тип данных, значения которого содержат компоненты, например массивы, структуры;
  - C) тип данных, значения которого являются массивами;
  - D) тип данных, значения которого являются целыми числами;
3. Сколько вызовов системных функций всегда будет в программе, независимо от ее реализации?
- A) всегда 2;
  - B) по крайней мере один;
  - C) не менее двух;
4. Сколько вызовов системных функций всегда будет в программе, независимо от ее реализации?
- A) всегда 2;
  - B) по крайней мере один;
  - C) не менее двух;
5. Что должно стать результатом анализа создаваемой системы?
- A) описание архитектуры;
  - B) прототипы системы;
  - C) выработка общих тактических приемов;
  - D) описание назначения системы;
6. Что должно стать результатом анализа создаваемой системы?
- A) описание архитектуры;
  - B) прототипы системы;
  - C) выработка общих тактических приемов;
  - D) описание назначения системы;
7. Каким типом будете пользоваться для хранения и обработки данных о количестве членов вашей семьи?
- A) short;
  - B) float;
  - C) long;
8. Каким типом будете пользоваться для хранения и обработки данных о количестве членов вашей семьи?
- A) short;
  - B) float;
  - C) long;
9. Что понимается под простым типом?
- A) тип данных, значения которого не содержат компонент, например, целое или символ, в отличии от массива или структуры;
  - B) тип данных, значения которого содержат компоненты, например массивы, структуры;
  - C) тип данных, значения которого являются массивами;
  - D) тип данных, значения которого являются целыми числами;
10. Что понимается под простым типом?
- A) тип данных, значения которого не содержат компонент, например, целое или символ, в отличии от массива или структуры;
  - B) тип данных, значения которого содержат компоненты, например массивы, структуры;

- C) тип данных, значения которого являются массивами;
- D) тип данных, значения которого являются целыми числами;

11. Что понимается под модульным программированием?

- A) организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия;
- B) организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия, описания интерфейса и описания реализации;
- C) организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия, простейшим модулем является функция;
- D) организация программы в виде совокупности строк с соблюдением правил их взаимодействия;

12. Что понимается под модульным программированием?

- A) организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия;
- B) организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия, описания интерфейса и описания реализации;
- C) организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия, простейшим модулем является функция;
- D) организация программы в виде совокупности строк с соблюдением правил их взаимодействия;

13. Чем должно завершаться определение функции, возвращающей значение?

- A) определение функции, возвращающей значения, обязано завершаться оператором return;
- B) определение функции, возвращающей значения, обязано завершаться оператором break;
- C) определение функции, возвращающей значения должно иметь пустое тело;
- D) определение функции, возвращающей значения должно иметь оператор перехода;

14. Чем должно завершаться определение функции, возвращающей значение?

- A) определение функции, возвращающей значения, обязано завершаться оператором return;
- B) определение функции, возвращающей значения, обязано завершаться оператором break;
- C) определение функции, возвращающей значения должно иметь пустое тело;
- D) определение функции, возвращающей значения должно иметь оператор перехода;

15. Функции, которые вызывают сами себя, называются

- A) динамическими;
- B) рекурсивными;
- C) априорными;

16. Функции, которые вызывают сами себя, называются

- A) динамическими;
- B) рекурсивными;
- C) априорными;

17. Параметр-ссылка является

- A) именем;
- B) адресом;
- C) значением;

18. Параметр-ссылка является

- A) именем;
- B) адресом;
- C) значением;

19. Если перед именем формального параметра в заголовке функции указан только тип, такой параметр называют

- A) параметром-аргументом;

- В) параметром-значением;
  - С) параметром-спецификатором;
20. Если перед именем формального параметра в заголовке функции указан только тип, такой параметр называют
- А) параметром-аргументом;
  - В) параметром-значением;
  - С) параметром-спецификатором;
21. Оператор return содержит
- А) значение функции;
  - В) параметры стека;
  - С) массив переменных;
22. Оператор return содержит
- А) значение функции;
  - В) параметры стека;
  - С) массив переменных;
23. К чему может привести общедоступность глобальных переменных?
- А) к переполнению стека;
  - В) к несогласованному их изменению разными функциями;
  - С) к невозможности контекстного вывода значений (динамических и статических);
24. К чему может привести общедоступность глобальных переменных?
- А) к переполнению стека;
  - В) к несогласованному их изменению разными функциями;
  - С) к невозможности контекстного вывода значений (динамических и статических);
25. Чем не может являться операция: << ?
- А) операция вывода;
  - В) сдвиг влево;
  - С) меньше или равно;
26. Чем не может являться операция: << ?
- А) операция вывода;
  - В) сдвиг влево;
  - С) меньше или равно;
27. Какие значения могут принимать переменные, объявленные следующим образом: int a,b;?
- А) целые;
  - В) вещественные;
  - С) значимые;
28. Какие значения могут принимать переменные, объявленные следующим образом: int a,b;?
- А) целые;
  - В) вещественные;
  - С) значимые;
29. Правильность вызова системных функций производится компилятором с помощью
- А) данных заголовочных файлов;
  - В) параметров первичной инициализации системы;
  - С) шаблонов потоковых данных;
30. Правильность вызова системных функций производится компилятором с помощью
- А) данных заголовочных файлов;
  - В) параметров первичной инициализации системы;

С) шаблонов потоковых данных;

31. Каждая строка программы, содержащая какое-либо объявление или выполняемое действие, оканчивается

- А) точкой;
- В) системным комментарием;
- С) точкой с запятой;

32. Каждая строка программы, содержащая какое-либо объявление или выполняемое действие, оканчивается

- А) точкой;
- В) системным комментарием;
- С) точкой с запятой;

## Примеры комплектов задач

1. Создайте консольное приложение, спрашивающее у пользователя имя и отвечающая в ответ, что приложение под авторством [ФИО автора] приветствует [Имя пользователя].
2. Создайте консольное приложение, запрашивающее у пользователя 2 числа и арифметический знак операции между ними и выводящее результат этой операции.
3. Создайте консольное приложение, запрашивающее у пользователя размер одномерного целочисленного массива, генерирующее одномерный массив случайным образом и выводящее элементы этого массива большие среднего арифметического.
4. Создайте консольное приложение для чтения файлов в формате csv (формат задается самостоятельно).
5. Создайте графическое приложение с четырьмя кнопками, на каждой кнопке должно быть указано одно из названий элементов из панели элементов, по нажатию на кнопку приложение должно выдать сообщение с кратким описанием названного элемента.
6. Создайте приложение, в котором будет строка меню внизу окна, 3 кнопки справа в серой колонке и в оставшемся пространстве 4 прямоугольника по 2 в каждой строке, Строка статуса
7. Создать приложение-окно с градиентной заливкой и с вертикальным и горизонтальным скроллбарами, в зависимости от состояний скроллбаров менять градиент
8. Создать приложение-окно с градиентной заливкой и, в зависимости от положения курсора мыши в окне менять градиент

## 1.2. Промежуточная аттестация

Промежуточная аттестация осуществляется в форме экзамена

В качестве средств, используемых на промежуточной аттестации предусматривается:

Билеты

## 1.3. Типовые задания для проведения промежуточной аттестации

Вопросы к экзамену

1. Язык программирования C#. Общая характеристика языка: место в классификации языков, синтаксис и семантика языка, основные объекты языка, операторы, структура программы.
2. Понятие среды программирования. Компоненты среды. Компиляция и компоновка программы.
3. Синтаксические и логические ошибки. Тестирование и отладка программы.
4. Данные в языке C#: константы и переменные. Скалярные типы данных. Модификаторы типов.
5. Данные числовых типов в языке C#: объявление, характеристика, допустимые операции, приведение типов. Пример использования.
6. Операции языка C#. Приоритет операций. Оператор и операция присваивания в языке C#. Множественное присваивание. Выражения.
7. Функции форматного ввода и вывода. Параметры. Управляющая строка. Спецификаторы формата. Управляющие символы.
8. Алгоритмическая конструкция ветвления: полная и неполная форма, блок-схемы. Условная операция. Условный оператор в языке C#: структура оператора, полная и неполная формы, использование сложных условий. Пример на языке C#.
9. Алгоритмическая конструкция выбора: понятие, блок-схема. Оператор выбора в языке C#: структура оператора. Пример программы на C#.
10. Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с предусловием (понятие, использование, блок-схема). Оператор цикла for: структура оператора, пример использования.
11. Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с предусловием (понятие, использование, блок-схема). Оператор цикла с предусловием while в языке C#: структура оператора, допустимые и недопустимые условия, пример использования.
12. Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с постусловием (понятие, использование, блок-схема). Оператор цикла с постусловием в языке C#: структура оператора, допустимые и недопустимые условия, пример использования.
13. Типовые циклические алгоритмы: максимум/минимум, сумма/произведение, количество.
14. Линейный массив: понятие массива, объявление, инициализация массива, индексация элементов. Формирование и вывод массива.
15. Указатели. Понятие указателя, объявление, инициализация. Операции, применимые к указателям. Указатель на указатель.
16. Использование указателей при работе с массивами. Получение адресов и значений элементов. Последовательный перебор элементов. Примеры.
17. Функции в языке C#: понятие, объявление, прототипы функций. Вызов функции. Типы возвращаемых значений.
18. Функции в языке C#: параметры формальные и фактические, механизм передачи параметров. Передача параметров «по значению» и «по ссылке». Пример использования.

19. Локальные и глобальные переменные. Области действия и области видимости. Экранирование переменных.
20. Библиотечные функции. Заголовочные файлы. Подключение библиотек. Функции математической библиотеки.
21. Массив символов и строка в языке C#. Ввод и вывод строк. Простейшие алгоритмы сканирования и обработки строки. Пример.
22. Строка в языке C#: библиотечные функции для обработки строк. Примеры использования.
23. Основные структурные части ИСР C#.
24. Основные проектные операции C#.
25. Назначение основных элементов знатока кода C#.
26. Постройте иерархию базовых классов VCL?
27. Перечислить и описать компоненты отображения информации.
28. Опишите особенности работы с канвой. Укажите компоненты, обладающие канвой.
29. Перечислите основные методы канвы.
30. Перечислите компоненты VCL, реализующие работу системных диалогов различного назначения. Укажите их основные свойства и методы.
31. Укажите основные принципы работы с приложением, окном, страницей.
32. Перечислите основные события окна и полотна.
33. Язык C#. Назначение класса исключений.
34. Язык C#. Классификация стандартных функций.
35. Приложение стандарта WPF
36. Установка и настройка приложения по стандарту WPF.



## 2. КРИТЕРИИ ОЦЕНКИ ПО ДИСЦИПЛИНЕ

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине				
Оценка	2 (не зачтено)	3 (зачтено)	4 (зачтено)	5 (зачтено)
виды оценочных средств				
<b>Знания</b> (виды оценочных средств: приведены в п. 1.2.)	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
<b>Умения</b> (виды оценочных средств: приведены в п. 1.2.)	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
<b>Навыки</b> (владения, опыт деятельности) (виды оценочных средств: приведены в п. 1.2..)	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач