

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЛГПУ»)**

Структурное подразделение Институт физико-математического
образования, информационных и обслуживающих технологий
Кафедра информационных образовательных технологий и систем

УТВЕРЖДАЮ

Врио директора ИФМОИОТ

Е.А. Журавлёва

« 15 » января 20 25 г.

Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
«Основы программирования»

По направлению подготовки 44.03.01 Педагогическое образование

Профиль подготовки Компьютерные системы и образовательная
робототехника

Квалификация выпускника – бакалавр

Форма обучения очная

Курс ОФО – 1 курс

Разработчик
канд. физ.-мат. наук, доцент кафедры
информационных образовательных
технологий и систем Швыров В.В.

заведующий кафедрой
информационных образовательных
технологий и систем
Д.А. Капустин

Протокол

от « 14 » января 20 25 г. № 9

Луганск, 2025

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Основы программирования и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины (модуля).

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО бакалавриат / специалитет / магистратура по направлению подготовки 44.03.01 Педагогическое образование, утвержденного приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 121 (с изменениями и дополнениями).

1.3. Перечень компетенций, формируемых в процессе освоения основной образовательной программы

Процесс освоения дисциплины направлен на формирование следующих компетенций и индикаторов их достижения:

Код по ФГОС ВО	Индикатор достижения
Универсальные	
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Демонстрирует знание особенностей системного и критического мышления, аргументированно формирует Собственное суждение и оценку информации, принимает обоснованное решение УК-1.2. Применяет логические формы и процедуры, способен к рефлексии по поводу Собственной и чужой мыслительной деятельности УК-1.3. Анализирует источники Информации с целью выявления их Противоречий и поиска достоверных суждений
Профессиональные	
ПК-4. Способен разрабатывать требования и проектировать программное обеспечение, а также программно-аппаратные комплексы	ПК-4.1. Знать: методы и технологию анализа и проектирования требований к программному обеспечению процессов и систем с заданной структурой и функциональными свойствами ПК-4.2. Уметь: осуществлять разработку требований и проектирование технических заданий на разработку программного обеспечения программно-аппаратных комплексов ПК-4.3. Владеть: навыками разработки программного обеспечения технологических процессов обучающей организации

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Стадии создания программного продукта	УК-1; ПК-4	Устный опрос
Тема 2. Определение Алгоритма. Свойства алгоритма	УК-1; ПК-4	Устный опрос
Тема 3. Синтаксис языка C++. Переменные и константы	УК-1; ПК-4	Выполнение лабораторных работ
Тема 4. Типы данных. Преобразование типов	УК-1; ПК-4	Выполнение лабораторных работ
Тема 5. Условный оператор	УК-1; ПК-4	Выполнение лабораторных работ
Тема 6. Оператор множественного выбора switch	УК-1; ПК-4	Выполнение лабораторных работ
Тема 7. Циклы с пред- и постусловием. Цикл for. Операторы break и continue	УК-1; ПК-4	Выполнение лабораторных работ
Тема 8. Одномерные массивы. Строки	УК-1; ПК-4	Выполнение лабораторных работ
Тема 9. Двумерные массивы. Массивы строк	УК-1; ПК-4	Выполнение лабораторных работ
Тема 10. Функции (Объявление, вызов, аргументы функции)	УК-1; ПК-4	Выполнение лабораторных работ
Тема 11. Прототипы, перегрузка, шаблоны функций	УК-1; ПК-4	Выполнение лабораторных работ
Тема 12. Сортировки	УК-1; ПК-4	Выполнение лабораторных работ
Тема 13. Поиск данных в массивах	УК-1; ПК-4	Выполнение лабораторных работ
Тема 14. Структуры. Массивы структур	УК-1; ПК-4	Выполнение лабораторных работ
Тема 15. Указатели. Ссылки	УК-1; ПК-4	Устный опрос
Тема 16. Динамическая память. Операторы new и delete	УК-1; ПК-4	Выполнение лабораторных работ
Тема 17. Одномерные и двумерные динамические массивы	УК-1; ПК-4	Выполнение лабораторных работ
Текущая аттестация	УК-1; ПК-4	Индивидуальная работа
Промежуточная аттестация	УК-1; ПК-4	Экзамен (письменный)

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
УК-1.	УК-1.1. Демонстрирует знание особенностей системного и критического мышления, аргументированно формирует собственное суждение и оценку информации, принимает обоснованное решение УК-1.2. Применяет логические формы и процедуры, способен к рефлексии по поводу Собственной и чужой мыслительной деятельности УК-1.3. Анализирует источники информации с целью выявления их противоречий и поиска достоверных суждений
ПК-4.	ПК-4.1. Знает: методы и технологию анализа и проектирования требований к программному обеспечению процессов и систем с заданной структурой и функциональными свойствами ПК-4.2. Умеет: осуществлять разработку требований и проектирование технических заданий на разработку программного обеспечения программно-аппаратных комплексов ПК-4.3. Владеет: навыками разработки программного обеспечения технологических процессов обучающей организации

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
	ОФО	О-ЗФО	ЗФО
Устные ответы на семинарских занятиях			
Выполнение и защита практических / лабораторных работ	45		
Самостоятельная работа	10		
Иные виды учебной работы (подготовка презентации, написание реферата, решение задач и др.)	10		
Экзамен	35		
Всего	100		

Накопительная система оценивания по 100-балльной шкале

Четырехбалльная система оценивания экзамена	100-балльная шкала	Буквенная шкала, соответствующая 100-балльной шкале	Система оценивания зачета
Отлично	90–100	А – отлично – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом сформированы; все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	
Хорошо	83–89	В – очень хорошо – теоретическое содержание курса освоено полностью, без пробелов;	

		необходимые практические навыки работы с освоенным материалом в основном сформированы; все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	Зачтено
Хорошо	75–82	С – хорошо – теоретическое содержание курса освоено полностью; некоторые практические навыки работы с освоенным материалом сформированы недостаточно; все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
Удовлетворительно	63–74	Д – удовлетворительно – теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера; необходимые практические навыки работы с освоенным материалом в основном сформированы; большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, содержат ошибки	
Удовлетворительно	50–62	Е – посредственно – теоретическое содержание курса освоено частично; некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	
Неудовлетворительно	21–49	ФХ – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	Не зачтено
Неудовлетворительно	0–20	Ф – неудовлетворительно – теоретическое содержание курса не освоено; необходимые практические навыки работы не сформированы; все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий	

2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

2.1. Оценочные средства текущего контроля

1. Стадии создания программного продукта.
2. Синтаксис языка C++ (Элементы и структура языка).
3. Переменные локальные и глобальные.
4. Константы
5. Библиотеки в C++.
6. Типы данных C++.
7. Преобразование типов
8. Условный оператор if else
9. Оператор множественного выбора switch
10. Циклы с пред- и постусловием while, do while.
11. Цикл for.
12. Операторы break и continue
13. Одномерные массивы. Назначение
14. Двумерные массивы. Назначение
15. Строки. Библиотеки работы со строками.
16. Массивы строк
17. Функции (Объявление, вызов, типы)
18. Аргументы функции
19. Прототипы функций
20. Перегрузка функций
21. Шаблоны функций
22. Структуры. Назначение, синтаксис.
23. Массивы структур.
24. Перечисления
25. Функции для работы с файлами
26. Указатели в C++.
27. Ссылки
28. Указатели и функции
29. Динамическая память.
30. Динамические массивы

Типовые задания по дисциплине

Задание 1. Написать программу.

1.1. Работник зарабатывает X руб. за каждый из 38 часов своей работы. Ему платят в 1.5 раза больше за каждый час сверх 38 часов. Какую сумму он получит, если отработает Y часов?

1.2. Даны 3 числа: X, Y, Z. Найдите произведение среднего и большего из них

1.3. Даны 2 числа a и b. Большее из этих чисел увеличьте в 4 раза, а меньшее – в 2 раза.

Задание 1. Написать программу. Вывести в значения в виде таблицы: x | y. Значение, t, s запрашиваются у пользователя. Шаг изменения x задать самостоятельно.

№ вариан та	Функция	Интервал изменения аргумента	Количество вычисляемы х значений
-------------------	---------	------------------------------------	----------------------------------------

1	$y = \sqrt[5]{t+x^3}$	0..5	150
2	$y = t \cdot \frac{3 \cdot x^5}{\sqrt{1+e^{x^3}}}$	-3..3	200
3	$y = \frac{\sin^2(x^t)}{\sqrt{1+x^3}}$	0..4	250

Задание 1. Написать программу

№ варианта	Задача
1	В матрице размером 10x10 подсчитать сумму элементов в каждой строке
2	В матрице размером 5x5 поменять местами столбцы и строки, результат сохранить в другую матрицу
3	Найти минимальный элемент каждого столбца матрицы 10x10 и определить его адрес

Задание 1. Написать программу

№ варианта	Задача
1	Даны две целочисленные квадратные матрицы порядка 6. Найти последовательность из нулей и единиц b_1, \dots, b_6 такую, что $b_i=1$, когда количество отрицательных и неотрицательных элементов i -й строки первой матрицы совпадает соответственно с количеством отрицательных и неотрицательных элементов i -й строки второй матрицы
2	Дана действительная матрица размера $m \times n$. Определить числа b_1, \dots, b_m , равные суммам элементов строк
3	Дана действительная матрица размера $m \times n$. Определить числа b_1, \dots, b_m , равные значениям средних арифметических элементов строк

Задание 1. Написать функцию согласно заданному варианту, организовать ее вызов в главной функции программы.

№	Задача 1	Переменные
1	$F = \frac{a}{x+2} e^{-bx^2} + \ln(a+bx) + \frac{a+bx}{e^{-bx^2} \operatorname{tg}\left(\frac{x+2}{a}\right)};$	$a=1,13;$ $b=-0,17;$ $x=0,92.$
2	$y = e^{-at} \frac{t+\sqrt{t+a}}{t-\sqrt{t-b}} + \frac{\ln \sqrt{t+a}}{\ln \sqrt{t-b}} * tge^{-at};$	$a=2,7;$ $b=1,3;$ $t=2,5.$

3	$z = 2^{-x} \arctg(x+a) - 3^{-bx} \cos(x+b) + \frac{\ln\left(\frac{x+a}{x+b}\right)}{2^{-x} + 3^{-bx}};$	$a=1,12;$ $b=-0,14;$ $x=2,49.$
---	----------------------------------------------------------------------------------------------------------	--------------------------------------

Задание 1. Написать функцию согласно заданному варианту, организовать ее вызов в главной функции программы.

№	Задача 1.
1	Написать функцию для поиска среднего значения элементов массива целых чисел. Параметрами функции является размер массива, возвращаемым значением – среднее значение.
2	Написать функцию, которая меняет местами два элемента массива. Номера меняемых элементов – параметры функции. С ее помощью отсортировать массив методом пузырьков.
3	Написать функцию, которая принимает два числа и возвращает истину, если первое больше второго. Использовать ее для сортировки массива по убыванию методом пузырьков.

Задание 1. Написать программу

1. Дана строка, состоящая из символов латинского алфавита, разделенных пробелами (одним или несколькими). Преобразовать каждое слово в строке, удалив из него все вхождения первой буквы этого слова (количество пробелов между словами не изменять).
2. Дана строка, состоящая из символов латинского алфавита, разделенных пробелами (одним или несколькими). Определить количество слов, которые начинаются и заканчиваются одной и той же буквой.
3. Дана строка-предложение из символов латинского алфавита. Вывести самое короткое слово в предложении (если таких слов несколько, то вывести первое из них).

Задание 1. Написать программу

1. Дан массив, состоящий из n элементов. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.
2. Дан массив, состоящий из n элементов. Назовем *серией* группу подряд идущих одинаковых элементов, а *длиной серии* – количество этих элементов (длина серии не может быть равна 1). Заменить каждую серию, длина которой больше k , на один наименьший элемент массива. Если таких серий нет, то массив оставить без изменений.
3. Дан массив, состоящий из n элементов. Назовем *серией* группу подряд идущих одинаковых элементов, а *длиной серии* – количество этих элементов (длина серии не может быть равна 1). Преобразовать массив, увеличив первую серию наибольшей длины на один элемент.

Задание 1. Написать программу

1. Дана матрица размера $n \times m$. Поменять местами столбцы, содержащие минимальный и максимальный элементы матрицы.

2. Дана матрица размера $n \times m$. Поменять местами ее строки так, чтобы их максимальные элементы образовывали возрастающую последовательность.
3. Дана квадратная матрица порядка $2n + 1$. Зеркально отразить ее элементы относительно побочной диагонали матрицы.

2.2. Оценочные средства для промежуточной аттестации

Теоретические вопросы к экзамену

1. Какие стадии создания программного продукта вы знаете?
2. Синтаксис языка C++ (Элементы и структура языка)?
3. Какие типы данных существуют?
4. Что такое переменная в C++?
5. Что такое область видимости?
6. Что такое константа?
7. Как подключать библиотеки в программе?
8. Отличие глобальных и локальных переменных?
9. Как объявить переменную типа целое число (integer) в C++?
10. Какая разница между операторами "==" и "=" в C++?
11. Что такое условный оператор if в C++ и как он используется?
12. Что такое цикл for в C++ и как он работает?
13. Что такое массив в C++ и как его объявить?
14. Что такое массив в C++?
15. Как объявить массив в C++?
16. Как инициализировать одномерный массив при объявлении?
17. Как получить доступ к элементам одномерного массива?
18. Как изменить значение элемента одномерного массива?
19. Как определить размер одномерного массива?
20. Как передать одномерный массив в функцию?
21. Как вернуть одномерный массив из функции?
22. Что такое двумерный массив в C++?
23. Как объявить двумерный массив?
24. Как инициализировать двумерный массив при объявлении?
25. Как получить доступ к элементам двумерного массива?
26. Как изменить значение элемента двумерного массива?
27. Как определить размер двумерного массива?
28. Как передать двумерный массив в функцию?
29. Как вернуть двумерный массив из функции?
30. Как использовать вложенные циклы для работы с двумерными массивами?
31. Какие существуют методы сортировки двумерных массивов?
32. Какие существуют методы сортировки массивов?
33. Как использовать многомерные массивы?
34. Что такое указатель в C++ и как его использовать?
35. Как объявить функцию в C++?
36. Что такое прототип функции?
37. Что означает параметр по умолчанию в функции?
38. Зачем нужны константные параметры в функциях?
39. Как передаются параметры в функцию?
40. Каким образом в C++ можно осуществить ввод и вывод данных?
41. Что такое константа в C++ и как её объявить?
42. Что такое перегрузка функций в C++ и как она работает?
43. Какие стандартные библиотеки есть в C++ и для чего они используются?
44. Как объявить и инициализировать одномерный массив в C++?
45. Как получить доступ к элементу массива в C++?
46. Как совершить вставку элемента в середину одномерного массива?

47. Как удалить последний элемент в массиве?
48. Как объявить и использовать цикл for в C++?
49. Как объявить и использовать цикл while в C++?
50. Как объявить и использовать цикл do-while в C++?
51. Зачем нужен оператор break?
52. Зачем нужен оператор continue?
53. Как определить длину (количество элементов) массива в C++?
54. Как объявить и инициализировать двумерный массив (матрицу) в C++?
55. Как перебрать все элементы двумерного массива с помощью вложенных циклов в C++?
56. Как использовать условный оператор if-else в C++?
57. Как использовать условный оператор switch-case в C++?
58. Как использовать операторы сравнения (>, <, >=, <=, ==, !=) в условных выражениях в C++?
59. Что такое &&?
60. Что такое ||?
61. Что такое !?
62. Зачем нужны {} в программе?
63. Что такое алгоритм в программировании и какие алгоритмы вы знаете?
64. Как отсортировать массив элементов типа int в порядке возрастания с помощью алгоритма сортировки пузырьком в C++?
65. Как отсортировать массив элементов типа int в порядке возрастания с помощью алгоритма сортировки выбором в C++?
66. Зачем нужны функции в программе?
67. Как написать функцию?
68. Зачем нужен прототип функции?
69. Как вызвать функцию?
70. Что такое параметры функции?
71. Зачем нужен return?
72. Что означает void?
73. Что такое шаблоны функций в C++ и как они используются?
74. Как объявить и использовать шаблонную функцию в C++?
75. Какие преимущества использования шаблонных функций в C++?
76. Что такое рекурсия и как она используется в программировании?
77. Как написать рекурсивную функцию для вычисления факториала числа в C++?
78. Что такое передача параметров по значению и по ссылке в функциях C++?
79. Как создать и использовать функцию с переменным числом аргументов (variadic function) в C++?
80. Что такое структура (структура данных) в C++ и как она объявляется?
81. Как объявить переменную типа структуры и получить доступ к ее членам в C++?
82. Как получить доступ к элементам структуры?
83. Как изменить значение элемента структуры?
84. Как передать структуру в функцию?
85. Как вернуть структуру из функции?
86. Как использовать массивы структур?
87. Как использовать вложенные структуры?
88. Какие существуют методы сортировки структур?
89. Что такое перечисление (enum) в C++ и как оно объявляется?
90. Как использовать перечисление в C++?
91. Что такое объединение (union) в C++ и как оно объявляется?
92. Как использовать объединение в C++?
93. Что такое указатель (pointer) в C++ и как он объявляется?
94. Как получить адрес переменной в C++?
95. Как получить значение переменной, на которую указывает указатель, в C++?

96. Что такое ссылка (reference) в C++ и как она объявляется?
97. В чем разница между указателями и ссылками в C++?
98. Как объявить и инициализировать строку (массив символов) в C++?
99. Как получить длину строки в C++?
100. Как сконкатенировать две строки в C++?
101. Как скопировать одну строку в другую в C++?
102. Какие библиотеки работают со строками?
103. Как проверить, содержит ли строка определенный подстроку в C++?
104. Как найти первое вхождение символа или подстроки в строке в C++?
105. Как осуществить поиск всех вхождений символа или подстроки в строке в C++?
106. Что такое строка в C++?
107. Как объявить строку в C++?
108. Как получить доступ к символам строки?
109. Как изменить значение символа строки?
110. Как определить длину строки?
111. Как объединить две строки?
112. Как сравнить две строки?
113. Какие существуют методы поиска подстроки в строке?
114. Как использовать функции библиотеки <string>?
115. Что такое динамический массив (dynamic array) в C++ и как он создается?
116. Как добавить элемент в динамический массив в C++?
117. Как удалить элемент из динамического массива в C++?
118. Как изменить размер динамического массива в C++?
119. Что такое линейный поиск?
120. Что такое бинарный поиск?
121. Каким образом можно работать с файлами в C++?
122. Что такое вектор (vector) в C++ и как он отличается от обычного массива?
123. Как объявить и инициализировать вектор в C++?
124. Как добавить элемент в конец вектора в C++?
125. Как получить доступ к элементам вектора в C++?
126. Как удалить элемент из вектора в C++?
127. Что такое указатель на функцию в C++?
128. Как объявить указатель на функцию в C++?
129. Что такое множество (set) в C++ и как оно работает?
130. Как объявить и инициализировать множество в C++?
131. Как добавить элемент в множество в C++?
132. Как удалить элемент из множества в C++?
133. Что такое ассоциативный массив (map) в C++ и как он работает?
134. Как объявить и инициализировать ассоциативный массив в C++?
135. Как добавить элемент в ассоциативный массив в C++?
136. Как получить доступ к элементам ассоциативного массива в C++?
137. Зачем нужны итераторы?
138. Что такое итераторы?
139. Какие бывают итераторы?
140. Что такое лямбда -выражения и зачем их использовать?

Типовые задания к экзамену

1. Заполнить массив n целыми числами, n – количество, вводит пользователь. Числа задаются случайным образом. Написать функцию, позволяющую добавлять новое число, введенное пользователем, на указанную пользователем позицию. Вывести результат.

2. Заполнить массив n вещественными числами, n – количество, вводит пользователь. Сами числа вводить с клавиатуры. Написать функцию, позволяющую удалять число с заданной пользователем позиции. Вывести результат.
3. Заполнить массив n буквами латинского алфавита, n – количество, вводит пользователь. Написать функцию, позволяющую извлекать гласные буквы в другой массив без повторений. Вывести результат.
4. Заполнить массив n строками, n – количество, вводит пользователь. Написать функцию, позволяющую перевернуть каждую строку задом наперед. Вывести результат.
5. Заполнить массив n целыми числами, числа задаются случайным образом. n – количество, вводит пользователь. Написать функцию, позволяющую заменить все числа меньше среднего арифметического этого массива на 0. Вывести результат.
6. Заполнить массив n вещественными числами, вводятся с клавиатуры. n – количество, вводит пользователь. Написать функцию, позволяющую заменить все числа больше среднего арифметического этого массива на него. Вывести результат.
7. Заполнить массив n целыми числами, числа задаются случайным образом. n – количество, вводит пользователь. Написать функцию, позволяющую заменить все четные числа на их сумму. Отсортировать массив. Вывести результат.
8. Заполнить массив n вещественными числами, вводятся с клавиатуры. n – количество, вводит пользователь. Написать функцию, позволяющую сформировать новый массив, элементы которого это суммы первого и последнего, второго и предпоследнего и т.д. элементов из первого массива. Вывести результат на экран.
9. Заполнить массив n символами. n – количество, вводит пользователь. Написать функцию, позволяющую извлекать символы-буквы латиницы в другой массив. Вывести результат.
10. Заполнить массив n строками. n – количество, вводит пользователь. Написать функцию, позволяющую сформировать из этого массива массив - строки, которого содержат цифры. Вывести результат.
11. Заполнить массив n строками. n – количество, вводит пользователь. Написать функцию, позволяющую сформировать из этого массива массив – строки, которого содержат цифры. Вывести результат.