

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

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

УТВЕРЖДАЮ

Бриго директора ИФМОИОТ

Е.А. Журавлёва

20__ г.



Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
Алгоритмы и структуры данных

По направлению подготовки 44.03.05 Педагогическое образование (с двумя
профилями подготовки)

Профиль подготовки «Технология. Информатика»

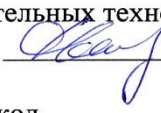
Квалификация выпускника бакалавр

Форма обучения очная/заочная

Курс очная – 1 курс, заочная – 1, 2 курс

Разработчик
ассистент Тивоненко А.А.

заведующий кафедрой информаци-
онных образовательных технологий и систем

 Д.А. Капустин

Протокол

от «13» сентября 2026 г. № 11

Луганск, 2026

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1 Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины «Алгоритмы и структуры данных» и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины.

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО бакалавриат по направлению подготовки 44.03.05 «Педагогическое образование (с двумя профилями подготовки)», утвержденным приказом Министерства науки и высшего образования Российской Федерации от 22.02.2018 г. № 125 (с изменениями и дополнениями от: 26 ноября 2020 г., 8 февраля 2021 г.) и Профессиональным стандартом, утвержденным Приказом Министерства труда и социальной защиты Российской Федерации «Об утверждении профессионального стандарта» от 8 сентября 2015 г. № 608 Н.

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.
Общепрофессиональные	
ОПК-8. Способен осуществлять педагогическую деятельность на основе специальных научных знаний	ОПК-8.1. Применяет методы анализа педагогической ситуации, профессиональной рефлексии на основе специальных научных знаний, в том числе в предметной области. ОПК-8.2. Проектирует и осуществляет учебно-воспитательный процесс с опорой на знания предметной области, психолого-педагогические знания и научно-обоснованные закономерности организации образовательного процесса.
Профессиональные	

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Алгоритмы виды и свойства	УК-1, ОПК-8	Устный опрос
Тема 2. Алгоритмы линейные, ветвящиеся, циклические	УК-1, ОПК-8	Устный опрос. Защита лабораторных работ
Тема 3. Одномерные и двумерные массивы	УК-1, ОПК-8	Устный опрос. Защита лабораторных работ
Тема 4. Алгоритмы рекурсивные	УК-1, ОПК-8	Защита лабораторных работ
Тема 5. Сортировки. Алгоритмы поиска	УК-1, ОПК-8	Защита лабораторных работ
Тема 6. Строки как тип данных. Алгоритмы работы со строками	УК-1, ОПК-8	Защита лабораторных работ
Тема 7. Пользовательские типы данных	УК-1, ОПК-8	Защита лабораторных работ
Текущая аттестация	УК-1, ОПК-8	Индивидуальное задание
Промежуточная аттестация	УК-1, ОПК-8	Зачет

1.5. Описание показателей формирования компетенций

Код по ФГОС ВО	Результаты сформированности
УК-1	Знает: как делать постановку задачи и определять основные пункты задачи при написании программы; какие задачи решаются с помощью программирования; как создавать алгоритмы, как писать код программы; как проводить отладку программы Умеет: ставить задачу, писать алгоритм к ней и реализовывать ее на языке программирования Владеет навыками: написания программ на языке программирования и решения поставленных задач с использованием программного обеспечения
ОПК-8	Знает: предметную область алгоритмизации и структурирования данных. Умеет: проектировать и реализовывать алгоритмические конструкции; обосновывать принимаемые проектные решения. Владеет навыками: объяснения использования составленных алгоритмов и применяемых структур данных

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		45
Самостоятельная работа	10		10

Иные виды учебной работы (подготовка презентации, написание реферата, решение задач и др.)	15		15
Зачет	30		30
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	Зачтено
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	
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 курса 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 учебной программой обучения учебные задания не выполнены либо качество выполненных некоторых из них оценено числом баллов, близким к минимальному	
Неудовлетворительно	21-49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы с освоенным материалом не сформированы; большинство предусмотренных учебной программой обучения учебных заданий не выполнено либо качество их выполнения оценено числом баллов, близким к минимальному;	Не зачтено

		при дополнительно самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	
Неудовлетворительно	0-20	F – 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

1.1. Оценочные средства текущего контроля

1. Виды алгоритмов, свойства алгоритмов, способы описания алгоритмов.
2. Оценка сложности алгоритмов.
3. Алгоритмы линейные, ветвящиеся, циклические
4. Одномерные массивы
5. Двумерные массивы
6. Строки. Библиотеки работы со строками.
7. Структуры. Назначение, синтаксис.
8. Массивы структур.
9. Перечисления
10. Объединения
11. Рекурсия
12. Алгоритмы поиска: линейный, бинарный
13. Сортировки. Методы вставки, выбором, быстрая, пузырьковая.

Типовые задания по дисциплине

1. Написать алгоритм и блок-схему, и программу входные параметры запросить у пользователя.

- 1.1. Введите два числа, обозначающие время и скорость, вычислите расстояние по введенным данным. Напишите обратные задачи (найти время, потом скорость).
- 1.2. Даны сила тока и сопротивление. Найти напряжение и вывести результат на экран. Сделать обратные задачи (найти сопротивление и силу тока по двум известным).
- 1.3. Составить программу вычисления длины окружности и площади круга по данному радиусу.

1. Написать программу, блок-схему к программе.

- 1.1. Даны три действительных числа x, y, z . Составить программу, вычисляющую $\min(x, y+z, xz)$.
- 1.2. Даны два действительных числа x, y . Составить программу, вычисляющую
$$z = \begin{cases} x \times y, & \text{если } x > 0, y > 0 \\ -y - x + 1 & \text{в противном случае} \end{cases}$$
.
- 1.3. Даны три действительных числа x, y, z . Составить программу, определяющую, существует ли треугольник со сторонами x, y, z .

1. Запросить у пользователя натуральное число. Выполнить задание согласно варианту. Написать блок-схему к программе.

- 1.1. Дано натуральное n . Составить блок-схему, вычисляющую количество цифр числа n .
- 1.2. Дано натуральное n . Составить блок-схему, вычисляющую первую цифру числа n .
- 1.3. Дано натуральное n . Составить блок-схему, вычисляющую произведение нечетных цифр числа n .

Задание 1 Написать блок-схему, алгоритм и программу

- 1.1. Запросить у пользователя массив из 10 целых элементов, вывести его на экран, вывести каждый элемент в квадрат и снова вывести на экран.
- 1.2. Найти максимальное число из всех четных элементов массива
- 1.3. Вычислить среднее арифметическое дробного массива размером 15. Значения элементов запрашиваются у пользователя

Задание 1 Написать программы и блок-схемы к программам

- 1.1. Запросить у пользователя 2 строки – имя и фамилию. Потом объединить их в одну и вывести на экран
- 1.2. Запрашивать у пользователя строки до тех пор, пока он не введет строку «Vse»;
- 1.3. Запросить у пользователя в цикле 10 строк и посчитать сумму символов в введенных строках

Задание 1 Написать программы

- 1.1. В массиве размером 20 отсортировать методом пузырьков элементы с каждым четным адресом
- 1.2. В массиве размером 20 отсортировать все нечетные элементы по возрастанию методом вставок
- 1.3. В двумерной матрице отсортировать строки по убыванию методом пузырьков.

1.2. Оценочные средства для промежуточной аттестации

1. Какие основные свойства должны иметь хорошие алгоритмы?
2. Какие основные типы алгоритмов существуют?
3. Что такое алгоритм в программировании и информатике?
4. Оценка сложности алгоритма?
5. Что такое алгоритм в программировании и какие типы алгоритмы вы знаете?
6. Какие свойства алгоритма вы знаете?
7. Что такое линейный алгоритм и как он выполняется?
8. Что такое условный алгоритм и как он выполняется?
9. Что такое циклический алгоритм и как он выполняется?
10. Какие алгоритмы сортировки существуют в C++?
11. Как работает алгоритм сортировки пузырьком в C++?
12. Как работает алгоритм сортировки выбором в C++?
13. Как работает алгоритм сортировки вставками в C++?
14. Какие алгоритмы поиска существуют в C++?
15. Как работает алгоритм линейного поиска в C++?
16. Как работает алгоритм бинарного поиска в C++?
17. Что такое блок-схема (flowchart) в программировании?
18. Какие элементы используются в блок-схемах и как они представляются?
19. Какие типы блоков (элементов) могут быть использованы в блок-схемах?
20. Какие символы используются для представления различных типов блоков в блок-схемах?
21. Какие виды соединений между блоками могут быть использованы в блок-схемах?
22. Какие основные правила следует учитывать при создании блок-схем?
23. Какие преимущества предоставляют блок-схемы при разработке программ?
24. Какие основные шаги нужно выполнить для создания блок-схемы?
25. Как читать и интерпретировать блок-схему?
26. Какие инструменты и программы можно использовать для создания блок-схем?
27. Какие типичные ошибки можно совершить при создании блок-схем и как их избежать?
28. Как блок-схемы связаны с процессом программирования и разработки алгоритмов?
29. Как использование блок-схем может помочь в обучении программированию и алгоритмическому мышлению?
30. Какие типы данных существуют?
31. Как происходит преобразование типов?
32. Что такое переменная в C++?
33. Что такое область видимости?

34. Отличие глобальных и локальных переменных?
35. Что такое одномерный массив в C++ и как его объявить?
36. Как объявить и инициализировать одномерный массив с элементами типа `int` в C++?
37. Как получить доступ к элементу массива в C++?
38. Как объявить и инициализировать двумерный массив (матрицу) в C++?
39. Как перебрать все элементы двумерного массива с помощью вложенных циклов в C++?
40. Что такое структура (структура данных) в C++ и как она объявляется?
41. Как объявить переменную типа структуры и получить доступ к ее членам в C++?
42. Что такое перечисление (`enum`) в C++ и как оно объявляется?
43. Как использовать перечисление в C++?
44. Что такое объединение (`union`) в C++ и как оно объявляется?
45. Как использовать объединение в C++?
46. Как объявить и инициализировать строку (массив символов) в C++?
47. Как получить длину строки в C++?
48. Как сконкатенировать две строки в C++?
49. Как скопировать одну строку в другую в C++?
50. Как проверить, содержит ли строка определенный подстроку в C++?
51. Как найти первое вхождение символа или подстроки в строке в C++?
52. Как осуществить поиск всех вхождений символа или подстроки в строке в C++?
53. Что такое динамический массив (`dynamic array`) в C++ и как он создается?
54. Как добавить элемент в динамический массив в C++?
55. Как удалить элемент из динамического массива в C++?
56. Как изменить размер динамического массива в C++?
57. Что такое вектор (`vector`) в C++ и как он отличается от обычного массива?
58. Как объявить и инициализировать вектор в C++?
59. Как добавить элемент в конец вектора в C++?
60. Как получить доступ к элементам вектора в C++?
61. Как удалить элемент из вектора в C++?
62. Что такое множество (`set`) в C++ и как оно работает?
63. Как объявить и инициализировать множество в C++?
64. Как добавить элемент в множество в C++?
65. Как удалить элемент из множества в C++?
66. Что такое ассоциативный массив (`map`) в C++ и как он работает?
67. Как объявить и инициализировать ассоциативный массив в C++?
68. Как добавить элемент в ассоциативный массив в C++?
69. Как получить доступ к элементам ассоциативного массива в C++?
70. Что такое рекурсия и как она используется в программировании?