


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

Врио директора ИФМОИОТ

 Е.А. Журавлёва  
« 11 » января 20 26 г.

Приложение к рабочей программе учебной дисциплины

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

для проведения текущего контроля и промежуточной аттестации  
обучающихся по дисциплине

**Практикум по решению олимпиадных задач по информатике**

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

Профиль подготовки «Математика. Информатика»

Квалификация выпускника бакалавр

Форма обучения очная/заочная

Курс 5

Разработчик  
ассистент Тивоненко А.А.

заведующий кафедрой информацион-  
ных образовательных технологий и систем  
Д.А. Капустин

Протокол

от « 13 » января 20 26 г. № 11

Луганск, 2026

# 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

### 1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины «Практикум по решению олимпиадных задач по информатике» 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) и Профессиональным стандартом, утвержденным Приказом Министерства труда и социальной защиты Российской Федерации «Об утверждении профессионального стандарта "Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)"» от 18 октября 2013 г. № 544н.

### 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	
Общепрофессиональные	
ОПК-2. Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)	ОПК.2.1. Осуществляет разработку программ отдельных учебных предметов, в том числе программ дополнительного образования (согласно освоенному профилю (профилям) подготовки)
	ОПК.2.2. Демонстрирует умение разрабатывать программу развития универсальных учебных действий средствами преподаваемой(-ых) учебных дисциплин, в том числе с использованием ИКТ
	ОПК.2.3. Демонстрируем умение разрабатывать планируемые результаты обучения и системы их оценивания, в том числе с использованием ИКТ (согласно освоенному профилю (профилям) подготовки)
Профессиональные	
ПК-3. Способен осваивать и применять	ПК.3.1. Способен формировать и

базовые научно-теоретические знания и практические умения по информатике в профессиональной деятельности.	реализовывать программы развития универсальных учебных действий по информатике ПК.3.2. Демонстрирует знание содержания образовательных программ по информатике ПК.3.3. Способен проектировать образовательные программы различных уровней и элементы образовательных программ в предметной области «Информатика»
---	--

#### 1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Организация олимпиад по информатике	ОПК-2 ПК-3	Устный опрос
Тема 2. Методика подготовки к олимпиадам по информатике	ОПК-2 ПК-3	Устный опрос
Тема 3. Особенности подготовки задач для олимпиад по информатике	ОПК-2 ПК-3	Устный опрос
Тема 4. Обзор типовых алгоритмов решения олимпиадных задач	ОПК-2 ПК-3	Выполнение лабораторных работ
Текущая аттестация	ОПК-2 ПК-3	Индивидуальное задание
Промежуточная аттестация	ОПК-2 ПК-3	Экзамен

#### 1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
	Знает: Умеет: Владеет навыками:
ОПК-2	Знает: классификацию олимпиад по информатике; нормативно-правовую базу организации олимпиад по информатике; системы автоматического проведения турниров; особенности заданий для олимпиад по информатике; типовые алгоритмы решения олимпиадных задач; методические особенности подготовки к решению олимпиадных задач. Умеет: анализировать существующие алгоритмы с точки зрения их эффективности и применимости для решения прикладных задач; решать типовые олимпиадные задачи; оценивать сложность разработанных алгоритмов и обосновывать их корректность. Владеет навыками: применять основы информатики и программи-

	рования к проектированию, конструированию и тестированию программных продуктов; навыками подготовки заданий и проведения олимпиад.
ПК-3	<p>Знает: синтаксис языка программирования – основные понятия и конструкции языков программирования (процедуры, функции, указатели), принципы написания программы.</p> <p>Умеет: использовать прикладное программное обеспечение общего и целевого назначения, в частности VisualStudio 2022 и выше, умеет производить отладку программ</p> <p>Владеет навыками: методами составления, описания и реализации алгоритмов; принципами и методами написания программ; принципами отладки программ</p>

### 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
Экзамен	30		30
<b>Всего</b>	<b>100</b>		

#### 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	<b>63-74</b>	<b>D</b> – 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	<b>50-62</b>	<b>E</b> – 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	<b>21-49</b>	<b>FX</b> – 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; при дополнительно самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	Не зачтено
Неудовлетворительно	<b>0-20</b>	<b>F</b> – 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. Виды алгоритмов, свойства алгоритмов, способы описания алгоритмов. Оценка сложности алгоритмов.
2. Нормативно-правовая база организации олимпиад по информатике.
3. Системы автоматизированного проведения турниров.
4. Основные методы, используемые для подготовки учащихся к олимпиадам.
5. Классификация олимпиадных задач по информатике.
6. Особенности олимпиадных задач. Требования к системе задач школьной олимпиады.
7. Подготовительный этап олимпиады. Проведение олимпиады. Проверка заданий, способы оценки. Подведение итогов.
8. Работа с системами автоматизированного проведения турниров по информатике
9. Типовые алгоритмы решения олимпиадных задач (задачи для начинающих)
10. Алгоритмы на дату и время для решения олимпиадных задач
11. Алгоритмы со строками решения олимпиадных задач
12. Задачи на формулы
13. Переборные алгоритмы
14. Алгоритмы на поиск
15. Сортировки в олимпиадных задачах
16. Жадные алгоритмы
17. комбинаторные алгоритмы
18. Задачи на бинарный поиск по ответу
19. Динамическое программирование
20. Задачи на поиск в ширину и поиск в глубину
21. Геометрические задачи в олимпиадном программировании

### **2.2 Оценочные средства для промежуточной аттестации**

1. Виды алгоритмов
2. Свойства алгоритмов
3. Способы описания алгоритмов.
4. Оценка сложности алгоритмов.
5. Что такое алгоритм в программировании и какие алгоритмы вы знаете?
6. Нормативно-правовая база организации олимпиад по информатике.
7. Системы автоматизированного проведения турниров.
8. Основные методы, используемые для подготовки учащихся к олимпиадам.
9. Классификация олимпиадных задач по информатике.
10. Особенности олимпиадных задач.
11. Требования к системе задач школьной олимпиады.
12. Подготовительный этап олимпиады.
13. Проведение олимпиады.
14. Проверка заданий, способы оценки.
15. Подведение итогов.
16. Работа с системами автоматизированного проведения турниров по информатике
17. Типовые алгоритмы решения олимпиадных задач (задачи для начинающих)
18. Алгоритмы на дату и время для решения олимпиадных задач
19. Алгоритмы со строками решения олимпиадных задач
20. Задачи на формулы
21. Переборные алгоритмы
22. Алгоритмы на поиск
23. Сортировки в олимпиадных задачах
24. Жадные алгоритмы

25. Комбинаторные алгоритмы
26. Задачи на бинарный поиск по ответу
27. Динамическое программирование
28. Задачи на поиск в ширину и поиск в глубину
29. Геометрические задачи в олимпиадном программировании
30. Что такое линейный поиск?
31. Что такое бинарный поиск?
32. Каким образом можно работать с файлами в C++?
33. Что такое вектор (vector) в C++ и как он отличается от обычного массива?
34. Что такое множество (set) в C++ и как оно работает?
35. Что такое ассоциативный массив (map) в C++ и как он работает?
36. Зачем нужны итераторы?
37. Как объявить и инициализировать строку (массив символов) в C++?
38. Как получить длину строки в C++?
39. Как сконкатенировать две строки в C++?
40. Как скопировать одну строку в другую в C++?
41. Какие библиотеки работают со строками?
42. Как проверить, содержит ли строка определенный подстроку в C++?
43. Как найти первое вхождение символа или подстроки в строке в C++?
44. Как осуществить поиск всех вхождений символа или подстроки в строке в C++?
45. Что такое динамический массив (dynamic array) в C++ и как он создается?
46. Как добавить элемент в динамический массив в C++?
47. Как удалить элемент из динамического массива в C++?
48. Как изменить размер динамического массива в C++?
49. Что такое структура (структура данных) в C++ и как она объявляется?
50. Как объявить переменную типа структуры и получить доступ к ее членам в C++?
51. Как получить доступ к элементам структуры?
52. Как изменить значение элемента структуры?
53. Как передать структуру в функцию?
54. Как вернуть структуру из функции?
55. Как использовать массивы структур?
56. Как использовать вложенные структуры?
57. Какие существуют методы сортировки структур?
58. Что такое перечисление (enum) в C++ и как оно объявляется?
59. Как использовать перечисление в C++?
60. Что такое объединение (union) в C++ и как оно объявляется?
61. Как использовать объединение в C++?
62. Что такое указатель (pointer) в C++ и как он объявляется?
63. Как получить адрес переменной в C++?
64. Как получить значение переменной, на которую указывает указатель, в C++?
65. Что такое ссылка (reference) в C++ и как она объявляется?
66. Что такое условный оператор if в C++ и как он используется?
67. Что такое цикл for в C++ и как он работает?
68. Что такое массив в C++ и как его объявить?
69. Что такое указатель в C++ и как его использовать?
70. Как объявить функцию в C++?
71. Что такое прототип функции?
72. Что означает параметр по умолчанию в функции?
73. Зачем нужны константные параметры в функциях?
74. Как передаются параметры в функцию?
75. Каким образом в C++ можно осуществить ввод и вывод данных?
76. Что такое константа в C++ и как её объявить?
77. Что такое перегрузка функций в C++ и как она работает?
78. Какие стандартные библиотеки есть в C++ и для чего они используются?

79. Как объявить и инициализировать одномерный массив в C++?
80. Как получить доступ к элементу массива в C++?
81. Как подключать библиотеки в программе?
82. Отличие глобальных и локальных переменных?
83. Как объявить и использовать цикл for в C++?
84. Как объявить и использовать цикл while в C++?
85. Как объявить и использовать цикл do-while в C++?
86. Какие типы данных существуют?
87. Как происходит преобразование типов?
88. Что такое переменная в C++?
89. Что такое область видимости?
90. Отличие глобальных и локальных переменных?
91. Что такое одномерный массив в C++ и как его объявить?
92. Как объявить и инициализировать одномерный массив с элементами типа int в C++?
93. Как получить доступ к элементам одномерного массива?
94. Как изменить значение элемента одномерного массива?
95. Как определить размер одномерного массива?
96. Как передать одномерный массив в функцию?
97. Как вернуть одномерный массив из функции?
98. Что такое двумерный массив в C++?
99. Как объявить двумерный массив?
100. Как инициализировать двумерный массив при объявлении?
101. Как получить доступ к элементам двумерного массива?
102. Как изменить значение элемента двумерного массива?
103. Как определить размер двумерного массива?
104. Как передать двумерный массив в функцию?
105. Как вернуть двумерный массив из функции?
106. Как объявить и инициализировать двумерный массив (матрицу) в C++?
107. Как перебрать все элементы двумерного массива с помощью вложенных циклов в C++?
108. Что такое структура (структура данных) в C++ и как она объявляется?
109. Как объявить переменную типа структуры и получить доступ к ее членам в C++?
110. Как получить длину строки в C++?
111. Как скопировать одну строку в другую в C++?
112. Как проверить, содержит ли строка определенный подстроку в C++?
113. Как найти первое вхождение символа или подстроки в строке в C++?
114. Как сравнить две строки?
115. Какие существуют методы поиска подстроки в строке?
116. Как использовать функции библиотеки <string>?
117. Что такое динамический массив (dynamic array) в C++ и как он создается?
118. Как добавить элемент в динамический массив в C++?
119. Как удалить элемент из динамического массива в C++?
120. Как изменить размер динамического массива в C++?
121. Что такое линейный поиск?
122. Что такое бинарный поиск?
123. Каким образом можно работать с файлами в C++?
124. Как осуществить поиск всех вхождений символа или подстроки в строке в C++?
125. Как объявить и инициализировать вектор в C++?
126. Как добавить элемент в конец вектора в C++?
127. Как получить доступ к элементам вектора в C++?
128. Как удалить элемент из вектора в C++?
129. Как объявить и инициализировать множество в C++?
130. Как добавить элемент в множество в C++?
131. Как удалить элемент из множества в C++?



132. Что такое ассоциативный массив (map) в C++ и как он работает?
133. Как объявить и инициализировать ассоциативный массив в C++?
134. Как добавить элемент в ассоциативный массив в C++?
135. Как получить доступ к элементам ассоциативного массива в C++?
136. Что такое рекурсия и как она используется в программировании?
137. Зачем нужны итераторы?
138. Что такое итераторы?
139. Какие бывают итераторы?
140. Что такое лямбда -выражения и зачем их использовать?