

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. Журавлева  
«14» января 2026 г.



Приложение к рабочей программе учебной дисциплины

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
для проведения текущего контроля и промежуточной аттестации  
обучающихся по дисциплине  
«Технология разработки программного обеспечения»**

**По направлению подготовки** 44.04.01 Педагогическое образование  
**Профиль подготовки** Информатика и образовательная робототехника  
**Квалификация выпускника** – магистр  
**Форма обучения** очная, заочная  
**Курс** ОФО – 2 курс, ЗФО – 2 курс

Разработчик  
Суворова Е.Ю.  
канд. пед. наук, доцент кафедры  
информационных образовательных  
технологий и систем  
Заведующий кафедрой  
Д.А. Капустин  
Протокол от «14» января 2026 г. № 11

Луганск, 2026

# 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

## 1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Технология разработки программного обеспечения 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 44.04.01 Педагогическое образование, утвержденного приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 126 (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
ПК-2. Интеграция разработанного программного обеспечения	ПК-2.1. Знать современные технологии интеграции разработанного системного программного обеспечения ПК-2.2. Уметь осуществлять интеграцию разработанного системного программного обеспечения ПК-2.3. Владеть навыками интеграции разработанного системного программного обеспечения
УК-3. Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	УК-3.1. Знает технологии управления проектами на всех этапах его жизненного цикла УК-3.2. Умеет осуществлять практическое управление проектами на всех этапах его жизненного цикла УК-3.3. Имеет практический опыт управления проектами на всех этапах его жизненного цикла

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

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Понятие жизненного цикла	ПК-2; УК-3	Выполнение лабораторных работ
Тема 2. Программное обеспечение и его классификация	ПК-2; УК-3	Выполнение лабораторных работ
Тема 3. Стадии разработки ПО	ПК-2; УК-3	Выполнение лабораторных работ

Тема 4. Общая схема процесса создания ПО	ПК-2; УК-3	Выполнение лабораторных работ
Тема 5. Основы построения интерфейсов	ПК-2; УК-3	Выполнение лабораторных работ
Тема 6. Стандартизация и сертификация ПО	ПК-2; УК-3	Выполнение лабораторных работ
Тема 7. Нормативная база в области документирования ПО	ПК-2; УК-3	Выполнение лабораторных работ
Текущая аттестация	ПК-2; УК-3	Контрольная работа
Промежуточная аттестация	ПК-2; УК-3	Экзамен

### 1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ПК-2. Интеграция разработанного программного обеспечения	<p>Знает: модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; виды и варианты интеграционных решений; современные технологии и инструменты интеграции;</p> <p>Умеет: анализировать проектную и техническую документацию; использовать специализированные графические средства построения и анализа архитектуры программных продуктов; организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов;</p> <p>Владеет навыками: выполнять тестирование интеграции; организовывать постобработку данных; создавать классы-исключения на основе базовых классов; выполнять ручное и автоматизированное тестирование программного модуля</p>
УК-3. Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	<p>Знает: методы управления проектами; этапы жизненного цикла проекта; принципы управления проектами и особенности их применения для различных этапов жизненного цикла стандартных моделей.</p> <p>Умеет: осуществлять мониторинг хода реализации проекта, корректировать отклонения, вносить дополнительные изменения в план реализации проекта, уточнять зоны ответственности участников проекта.</p> <p>Владеет навыками: составления плана-графика реализации проекта в целом и плана-контроля его выполнения;</p> <p>конструктивного преодоления возникающих разногласий и конфликтов.</p>

### 1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов
--------------------	-------------------

3 семестр / 5-6 триместр			
	<b>ОФО</b>	<b>О-ЗФО</b>	<b>ЗФО</b>
Защита лабораторных работ	50 баллов		50 баллов
Выполнение заданий самостоятельной работы	30 баллов		30 баллов
экзамен	20 баллов		20 баллов
<b>Итого за семестр:</b>	100 баллов		100 баллов
<b>Всего</b>	100 баллов		

### Накопительная система оценивания по 100-балльной шкале

Четырехбалльная система оценивания экзамена	100-балльная шкала	Буквенная шкала, соответствующая 100-балльной шкале	Система оценивания зачета
Отлично	<b>90–100</b>	<b>А</b> – 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	<b>83–89</b>	<b>В</b> – 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	<b>75–82</b>	<b>С</b> – 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>Д</b> – 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	<b>50–62</b>	<b>Е</b> – 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	<b>21–49</b>	<b>FX</b> – 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	<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 (типовые)**

Вопросы для текущего контроля:

1. Что такое жизненный цикл разработки программного обеспечения (SDLC) и какие фазы включает?
2. Опишите основные принципы Agile методологий разработки ПО.
3. Какие роли выделяются в методологии Scrum и какие функции выполняют?
4. Чем отличается модель Waterfall от Agile подхода в разработке ПО?
5. Какие преимущества предоставляет использование Continuous Integration (CI) в разработке ПО?
6. Какие инструменты часто используются для реализации Continuous Delivery (CD)?
7. Что такое микросервисная архитектура и какие преимущества она предоставляет?
8. Какие принципы лежат в основе Test-Driven Development (TDD)?
9. Что такое контроль версий и зачем он используется в разработке ПО?
10. Какие основные принципы лежат в основе Git и какие команды чаще всего используются?
11. Какие методы обеспечения безопасности ПО существуют и как они применяются?
12. Что такое архитектурные атаки и как они могут повлиять на программное обеспечение?
13. Какие принципы следует придерживаться при разработке доступного программного обеспечения?
14. Какие методы тестирования масштабирования ПО существуют и как они проводятся?
15. Что такое отладка (debugging) и какие методы отладки ПО существуют?
16. Какие принципы следует придерживаться при разработке устойчивого к ошибкам программного обеспечения?
17. Какие методы мониторинга и логирования используются для обеспечения качества ПО?
18. Что такое интеграционное тестирование и как оно проводится в разработке ПО?
19. Какие инструменты часто используются для проведения модульного тестирования?
20. Чем отличается приёмочное тестирование от других видов тестирования и как оно проводится в разработке ПО?



## 2.2. Оценочные средства для промежуточной аттестации

Вопросы для проведения аттестации

1. Что такое жизненный цикл разработки программного обеспечения (SDLC)?
2. Какие основные фазы включает жизненный цикл разработки программного обеспечения?
3. Что такое Waterfall модель разработки ПО и какие её особенности?
4. В чём преимущества и недостатки Waterfall модели?
5. Что такое Agile методологии разработки ПО и какие её особенности?
6. Какие основные принципы лежат в основе Agile методологий?
7. Какие методы коммуникации предпочтительны в Agile разработке ПО?
8. Что такое Scrum и какие роли в нём выделяются?
9. Какие события происходят в рамках Scrum методологии?
10. Какие артефакты создаются в рамках Scrum процесса?
11. Какие основные принципы лежат в основе Kanban методологии разработки ПО?
12. Как работает Kanban-доска?
13. В чём разница между Scrum и Kanban методологиями?
14. Что такое DevOps и какие основные принципы он включает?
15. Какие инструменты часто используются в DevOps практиках?
16. Какие преимущества предоставляет DevOps в разработке ПО?
17. Что такое Continuous Integration (CI) и какие принципы он включает?
18. Какие инструменты часто используются для реализации CI?
19. Что такое Continuous Delivery (CD) и какие принципы он включает?
20. Какие инструменты часто используются для реализации CD?
21. Что такое контейнеризация и какие инструменты используются для неё?
22. Какие преимущества контейнеризации в разработке ПО?
23. Что такое микросервисная архитектура и какие преимущества она предоставляет?
24. Какие инструменты часто используются для управления микросервисами?
25. Какие принципы лежат в основе Domain-Driven Design (DDD)?
26. Что такое Test-Driven Development (TDD) и какие преимущества он предоставляет?
27. Какие шаги следует выполнить для реализации TDD?
28. Что такое Behavior-Driven Development (BDD) и какие преимущества он предоставляет?
29. Какие шаги следует выполнить для реализации BDD?
30. Какие инструменты часто используются для автоматизации тестирования в разработке ПО?

31. Что такое контроль версий и какие основные системы контроля версий существуют?
32. Какие преимущества предоставляет использование систем контроля версий в разработке ПО?
33. Какие основные принципы лежат в основе Git?
34. Какие команды Git часто используются в рамках разработки ПО?
35. Что такое ветвление и слияние (branching and merging) в Git?
36. Какие стратегии ветвления часто используются в Git?
37. Какие основные принципы лежат в основе GitHub Flow?
38. Как работает GitHub Flow?
39. Что такое контроль версий семантического уровня (Semantic Versioning) и какие принципы он включает?
40. Какие преимущества предоставляет использование Semantic Versioning в разработке ПО?
41. Какие основные этапы включает процесс разработки программного обеспечения?
42. Что такое требования к ПО и какие виды требований существуют?
43. Какие методы сбора требований к ПО существуют?
44. Что такое функциональные требования к ПО и как они отличаются от нефункциональных требований?
45. Как описываются требования к ПО в рамках Use Case моделирования?
46. Что такое User Stories и как они используются в Agile разработке ПО?
47. Какие атрибуты должны быть у хорошо описанного требования к ПО?
48. Что такое декомпозиция требований к ПО и какие методы декомпозиции существуют?
49. Какие преимущества предоставляет использование UML для моделирования требований к ПО?
50. Какие диаграммы UML чаще всего используются для моделирования требований к ПО?
51. Что такое проектирование архитектуры ПО и какие основные принципы оно включает?
52. Какие виды архитектурных стилей существуют в разработке ПО?
53. Что такое Monolithic архитектура и какие преимущества и недостатки она предоставляет?
54. Что такое Service-Oriented архитектура (SOA) и какие преимущества и недостатки она предоставляет?
55. Что такое Microservices архитектура и какие преимущества и недостатки она предоставляет?
56. Какие принципы лежат в основе RESTful архитектуры?
57. Что такое декомпозиция архитектуры ПО и какие методы декомпозиции существуют?
58. Какие диаграммы UML чаще всего используются для моделирования архитектуры ПО?



59. Что такое паттерны проектирования и какие основные категории паттернов существуют?
60. Какие принципы лежат в основе SOLID принципов объектно-ориентированного программирования?
61. Какие принципы лежат в основе GRASP принципов объектно-ориентированного проектирования?
62. Какие принципы лежат в основе DRY принципов программирования?
63. Что такое антипаттерны в программировании и какие примеры существуют?
64. Что такое рефакторинг кода и какие основные техники рефакторинга существуют?
65. Какие преимущества предоставляет использование атомарных коммитов в разработке ПО?
66. Какие основные принципы следует придерживаться при написании комментариев в коде?
67. Что такое код-ревью и какие преимущества он предоставляет в разработке ПО?
68. Какие принципы лежат в основе эффективного управления версиями ПО?
69. Что такое Continuous Integration (CI) и какие основные принципы он включает?
70. Какие преимущества предоставляет использование CI в разработке ПО?
71. Что такое Continuous Delivery (CD) и какие основные принципы он включает?
72. Какие преимущества предоставляет использование CD в разработке ПО?
73. Что такое Continuous Deployment и как оно отличается от Continuous Delivery?
74. Какие принципы следует придерживаться при разработке безопасного программного обеспечения?
75. Какие основные уязвимости безопасности ПО существуют и как их можно предотвратить?
76. Что такое аутентификация и авторизация в контексте разработки ПО?
77. Какие методы аутентификации и авторизации часто используются в разработке ПО?
78. Что такое шифрование данных и как оно применяется в разработке ПО?
79. Какие методы тестирования безопасности ПО существуют?
80. Какие методы обеспечения защиты ПО от вредоносного кода существуют?
81. Что такое архитектурные атаки и как они могут быть предотвращены?
82. Какие принципы следует придерживаться при разработке доступного программного обеспечения?

83. Какие методы тестирования доступности ПО существуют?
84. Какие принципы следует придерживаться при разработке масштабируемого программного обеспечения?
85. Какие методы тестирования масштабируемости ПО существуют?
86. Что такое производительность ПО и как её можно измерить?
87. Какие методы улучшения производительности ПО существуют?
88. Какие принципы следует придерживаться при разработке поддерживаемого программного обеспечения?
89. Какие методы обеспечения поддерживаемости ПО существуют?
90. Что такое отладка (debugging) и какие методы отладки ПО существуют?
91. Какие принципы следует придерживаться при разработке устойчивого к ошибкам программного обеспечения?
92. Какие методы тестирования устойчивости к ошибкам ПО существуют?
93. Что такое мониторинг и логирование в контексте разработки ПО?
94. Какие методы мониторинга и логирования часто используются в разработке ПО?
95. Что такое масштабирование программного обеспечения и какие методы масштабирования существуют?
96. Какие принципы следует придерживаться при разработке программного обеспечения с высокой доступностью?
97. Какие методы тестирования высокой доступности ПО существуют?
98. Что такое интеграционное тестирование и как оно проводится в разработке ПО?
99. Какие инструменты часто используются для проведения интеграционного тестирования?
100. Что такое модульное тестирование и как оно проводится в разработке ПО?
101. Какие инструменты часто используются для проведения модульного тестирования?
102. Что такое приёмочное тестирование и как оно проводится в разработке ПО?
103. Какие инструменты часто используются для проведения приёмочного тестирования?
104. Что такое функциональное тестирование и как оно проводится в разработке ПО?
105. Какие инструменты часто используются для проведения функционального тестирования?
106. Что такое нагрузочное тестирование и как оно проводится в разработке ПО?
107. Какие инструменты часто используются для проведения нагрузочного тестирования?
108. Что такое тестирование на проникновение и как оно проводится в разработке ПО?

109. Какие инструменты часто используются для проведения тестирования на проникновение?
110. Что такое тестирование безопасности и как оно проводится в разработке ПО?
111. Какие инструменты часто используются для проведения тестирования безопасности?
112. Что такое автоматизация тестирования и какие преимущества она предоставляет?
113. Какие инструменты часто используются для автоматизации тестирования?
114. Какие принципы следует придерживаться при выборе инструментов для автоматизации тестирования?
115. Что такое контроль качества ПО и какие основные методы контроля качества существуют?
116. Какие метрики часто используются для оценки качества ПО?
117. Что такое отчёты о качестве ПО и какие информации они должны содержать?
118. Какие инструменты часто используются для создания отчётов о качестве ПО?
119. Что такое CI/CD пайплайн и как он работает в разработке ПО?
120. Какие преимущества предоставляет использование CI/CD пайплайна в разработке ПО?
121. Что такое развертывание ПО и какие методы развертывания существуют?
122. Какие инструменты часто используются для автоматизации развертывания ПО?
123. Какие принципы следует придерживаться при развертывании ПО в production среду?
124. Что такое мониторинг ПО и какие инструменты используются для мониторинга?
125. Какие метрики часто мониторятся в процессе мониторинга ПО?
126. Что такое логирование ПО и какие инструменты используются для логирования?
127. Какие основные типы логов существуют и как они используются в разработке ПО?
128. Что такое анализ логов ПО и какие инструменты используются для анализа логов?
129. Какие методы анализа логов часто используются для выявления проблем в ПО?
130. Что такое мониторинг производительности ПО и как он проводится?
131. Какие метрики производительности часто мониторятся в процессе мониторинга производительности ПО?
132. Что такое масштабирование ПО и какие методы масштабирования существуют?
133. Какие инструменты часто используются для автоматизации масштабирования ПО?

134. Какие принципы следует придерживаться при разработке высокодоступного ПО?
135. Какие методы тестирования высокодоступности ПО существуют?
136. Что такое аудитория ПО и как она влияет на процесс разработки?
137. Какие методы сбора обратной связи от аудитории ПО существуют?
138. Какие принципы следует придерживаться при управлении изменениями в ПО?
139. Какие методы управления изменениями в ПО часто используются?
140. Какие принципы следует придерживаться при документировании ПО?