

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

«14» сентября 2026 г.

Приложение к рабочей программе учебной дисциплины

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

По направлению подготовки 09.03.04 Программная инженерия

Профиль подготовки Программное обеспечение систем и комплексов

Квалификация выпускника – бакалавр

Форма обучения очная

Курс ОФО – 4 курс

Разработчик

Капустин Д.А.

канд. техн. наук, доцент кафедры
информационных технологий и систем

Заведующий кафедрой

Д.А. Капустин

Протокол от «15» сентября 2026 г. № 14

Луганск, 2026

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Конструирование 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 по направлению подготовки 09.03.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 920 (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
Профессиональные	
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1. Знать основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ОПК-6.2. Уметь применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-6.3. Владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Основные понятия конструирования ПО.	ОПК-6	Выполнение лабораторных работ

Тема 2. Управление конструированием.	ОПК-6	Выполнение лабораторных работ
Тема 3. Подготовка к конструированию программного кода.	ОПК-6	Выполнение лабораторных работ
Тема 4. Качество программного кода	ОПК-6	Выполнение лабораторных работ
Тема 5. Совместное конструирование.	ОПК-6	Выполнение лабораторных работ
Текущая аттестация	ОПК-6	Контрольная работа
Промежуточная аттестация	ОПК-6	Зачет

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-6.3. Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
7 семестр			
	ОФО	О-ЗФО	ЗФО
Оформление отчетов по лабораторным работам	30 баллов	-	-
Работа на лабораторных занятиях	30 баллов	-	-
Выполнение тестовых заданий	-	-	-
Выполнение заданий самостоятельной работы	10 баллов	-	-
зачета	30 баллов	-	-
Итого за семестр:	100 баллов	-	-
Всего	100 баллов		

Накопительная система оценивания по 100-балльной шкале

Четырехбал- льная система оценивания экзамена	100- балльная шкала	Буквенная шкала, соответствующая 100- балльной шкале	Система оценивания зачета
Отлично	90–100	A – 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	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	75–82	C – 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	
Удовлетво- рительно	63–74	D – 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	
Удовлетво- рительно	50–62	E – 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	
Неудовлетво- рительно	21–49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество	Не зачтено

		их выполнения оценено числом баллов, близким к минимальному; при дополнительной 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

2.1. Оценочные средства текущего контроля (типовые)

Вопросы для текущего контроля:

1. Осуществляется ли правильная повторная инициализация переменных в коде, который выполняется более одного раза?
2. Все ли переменные имеют как можно меньшую область видимости?
3. Каждая ли переменная имеет одну и только одну цель?
4. Инициализировали ли вы переменные при их объявлении, если такое возможно?
5. В каждом ли методе проверяется корректность входных параметров?
6. Описывает ли имя представляемую переменной сущность полно и точно?
7. Спецификаторы вычисляемых значений находятся в конце имен?
8. Всем ли «временным» переменным присвоены выразительные имена?
9. Можно ли по именам булевых переменных понять, какой смысл имеют значения «истина» и «ложь»?
10. Удовлетворены ли вы тем, как выполнена декомпозиция классов на методы?
11. Достигнуто ли минимальное взаимодействие классов между собой?
12. Попробовали ли вы выполнить декомпозицию системы несколькими способами с целью нахождения наилучшего варианта?
13. Использовали ли вы для решения проблемы и нисходящий, и восходящий способы проектирования?
14. Довели ли вы проектирование до той точки, в которой реализация проекта кажется очевидной?
15. Имеет ли класс главную цель?
16. Удачное ли имя присвоено классу? Описывает ли оно главную цель класса?
17. Формирует ли интерфейс класса согласованную абстракцию?
18. Ясно ли интерфейс описывает использование класса?
19. Независим ли класс от других классов? Слабо ли он связан?
20. Сделаны ли члены класса минимально доступными?
21. Проверяли ли вы, нет ли дублирования кода, которое можно устранить?
22. Используете ли вы принцип DRY (Don't Repeat Yourself) в коде?
23. Проверяли ли вы, насколько легко тестировать ваш код?
24. Учитывали ли вы возможность расширения функциональности в будущем?

25. Проверяли ли вы, насколько легко читаем и понятен ваш код для других разработчиков?
26. Используете ли вы соглашения по именованию, принятые в вашей команде или языке программирования?
27. Проверяли ли вы, насколько эффективно используются ресурсы (память, процессор) в вашем коде?
28. Учитывали ли вы возможные ошибки и исключения при проектировании системы?
29. Проверяли ли вы, насколько легко поддерживать и изменять ваш код?
30. Используете ли вы шаблоны проектирования для решения типичных задач?
31. Проверяли ли вы, насколько безопасен ваш код с точки зрения уязвимостей?
32. Учитывали ли вы производительность кода при работе с большими объемами данных?
33. Проверяли ли вы, насколько хорошо документирован ваш код?
34. Используете ли вы инструменты статического анализа кода для поиска потенциальных проблем?
35. Проверяли ли вы, насколько хорошо ваш код соответствует принципам SOLID?

2.2. Оценочные средства для промежуточной аттестации

Вопросы для проведения аттестации

1. Понятие конструирования программных средств.
2. Место конструирования в жизненном цикле программного обеспечения.
3. Стандарт ГОСТ 34.601-90.
4. Стандарт ISO/IEC 12207:1995.
5. Стандарты и модели конструирования.
6. Модели жизненного цикла.
7. Классический (каскадный) жизненный цикл.
8. Основные принципы макетирования.
9. Какие этапы содержит инкрементная модель?
10. Быстрая разработка приложений.
11. Компонентно-ориентированная модель.
12. Анализ требований и определение спецификаций программного обеспечения.
13. Планирование конструирования.
14. Измерения в конструировании.
15. Проектирование в конструировании.

16. Методология SWEBOK.
17. Методология Custom Development Method (CDM).
18. Методология Rational Unified Process (RUP).
19. Методология Microsoft Solutions Framework (MSF).
20. Понятие тестирования ПО.
21. Какие существуют виды тестирования программного обеспечения?
22. Что такое модульное тестирование и как оно проводится?
23. Какие инструменты используются для автоматизации тестирования?
24. Что такое интеграционное тестирование и его основные подходы?
25. Какие существуют методы обеспечения качества программного обеспечения?
26. Что такое регрессионное тестирование и когда оно применяется?
27. Какие существуют подходы к тестированию пользовательского интерфейса?
28. Что такое нагрузочное тестирование и для чего оно используется?
29. Какие существуют стандарты и методики тестирования ПО?
30. Что такое тестирование безопасности и как оно проводится?
31. Какие существуют подходы к тестированию в agile-средах?
32. Что такое тестирование на основе сценариев использования?
33. Какие существуют методы оценки покрытия кода тестами?
34. Что такое тестирование совместимости и как оно проводится?
35. Какие существуют подходы к тестированию в распределенных системах?