МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВО Кыргызско-Российский Славянский университет имени первого Президента Российской Федерации Б. Н. Ельцина

Высшая школа Цифровых технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Конструирование ПО»

Уровень высшего образования

МАГИСТРАТУРА

Направление подготовки

09.04.04 - РФ, 710400 - КР Программная инженерия

(код и наименование направления подготовки)

"Основы анализа и разработки приложений с большими объемами распределенных данных"

(профиля образовательной программы)

Квалификация

магистр

Фонд оценочных средств предназначен для текущего, рубежного и промежуточного контроля результатов обучения магистрантов по дисциплине «Функциональное и логическое программирование» направления 09.04.04 «Программная инженерия».

Фонд оценочных средств рассмотрен и утвержден
на заседании Высшей школы Цифровых технологий протокол №1 от 02.09.26

Директор Высшей школы Цифровых технологий:

к.т.н., доцент Хмелева И.В.

Руководитель образовательной программы:

д.т.н., профессор Лыченко Н.М.

Исполнители (разработчики):

ст. преподаватель Беляев А.А.

СОГЛАСОВАНО:
Зам. директора по учебной работе
Политехнического института КРСУ Комарцов Н.М.

Раздел 1. Перечень компетенций с указанием этапов их формирования

Компетенция	Планируемые результаты обучения (по РПД)	Виды оценочных средств
ПК-4. Знать	Методы разработки программного обеспечения различного уровня сложности; понятия и методы конструирования ПО, предварительные условия, требования и архитектуру; проектирование, метрики, языки программирования, фреймворки, конвенции и методы управления сложностью.	Блок A: устные вопросы, контрольные работы № 1–4; блок D: теоретические вопросы зачёта.
ПК-4. Уметь	Декомпозировать задачи разработки на подзадачи; определять тип ПО и модель разработки; выбирать язык, фреймворки и методики конструирования; формулировать требования, оценивать предварительные работы, строить архитектуру и проектировать компоненты.	Блок B: аналитические и расчётно-проектные задания; блок C: практические работы № 1–11.
ПК-4. Владеть	Способами разработки и модернизации программного кода для компонентов информационных и автоматизированных систем; навыками архитектурного моделирования, UML, CRC-карточками, сравнением проектных альтернатив, документированием компонентов и контролем качества кода.	Блок C: практические решения и их защита; блок D: практическое задание и защита решения на зачёте.

Обозначения: блок A - репродуктивный уровень; блок B - реконструктивный уровень; блок C - практико-ориентированный и исследовательский уровень; блок D - промежуточная аттестация.

Раздел 2. Технологическая карта дисциплины

Курс 1, семестр 1; 4 ЗЕТ; 18 учебных недель; итоговая форма контроля - зачёт с оценкой.

Название модуля (раздела РПД)	Контроль	Форма контроля	Мин., балл	Макс., балл	Неделя
1. Основы конструирования программного обеспечения и подходы к разработке	Текущий	Защита практической работы № 1: применение методов и метафор разработки ПО.	3	6	4
	Рубежный	Контрольная работа № 1: принципы конструирования ПО, сравнение подходов и метафор.	3	6	4
2. Предварительные условия, требования и архитектура программных систем	Текущий	Защита практических работ № 2–5: тип ПО, требования, архитектура, метрики.	6	11	8
	Рубежный	Контрольная работа № 2: анализ предварительных условий, требований и архитектуры.	6	11	8
3. Выбор языков, технологий и методик конструирования программного обеспечения	Текущий	Защита практических работ № 6–9: язык, конвенции, фреймворки и процессы.	6	10	12
	Рубежный	Контрольная работа № 3: обоснованный выбор технологий и методик разработки.	6	10	12
4. Методы проектирования программных систем и компонентов	Текущий	Защита практических работ № 10–11: итеративное проектирование, UML и CRC.	5	8	16
	Рубежный	Контрольная работа № 4: управление сложностью и проектирование компонентов.	5	8	16
ВСЕГО за семестр			40	70	16

Промежуточная аттестация	Зачёт с оценкой	Теоретический вопрос, индивидуальное проектное задание, защита решения.	20	30	-
СЕМЕСТРОВЫЙ РЕЙТИНГ			60	100	-

Раздел 3. Типовые контрольные задания и оценочные материалы

Блок А. Теоретические вопросы для текущего контроля

Модуль 1. Контрольные вопросы

1. Что понимается под конструированием программного обеспечения и как оно связано с требованиями и проектированием?
2. В чём различие высокоуровневого и детального проектирования?
3. Какие метафоры применяются к процессу разработки программного обеспечения?
4. Чем отличаются «написание кода», «выращивание системы», «приращение» и «построение ПО» как метафоры?
5. Какие факторы проекта влияют на выбор метафоры и метода конструирования?
6. Как комбинируют разные подходы к разработке в рамках одного проекта?

Модуль 2. Контрольные вопросы

1. Что называют предварительными условиями конструирования и почему они влияют на качество ПО?
2. Какие признаки позволяют отнести проект к бизнес-системе, системе целевого назначения или критичной к качеству встроенной системе?
3. В чём отличие последовательного и итеративного процесса?
4. Как связаны проблема, видение проекта, требования и запросы на изменения?
5. Какие компоненты включает высокоуровневая архитектура программной системы?
6. Какие критерии применяют для принятия или отклонения запросов на изменение?
7. Какие метрики используют при оценке трудоёмкости предварительных работ?

Модуль 3. Контрольные вопросы

1. По каким критериям выбирают язык программирования для определённого типа ПО?
2. Что такое стандарт и конвенция кодирования? Как контролируют их выполнение?
3. Чем архитектурный шаблон отличается от программного фреймворка?
4. Какие факторы влияют на выбор библиотек и фреймворка?
5. Как выбирают процесс и методики конструирования в зависимости от ограничений проекта?
6. Как использовать инструмент статического анализа для контроля качества кода?

Модуль 4. Контрольные вопросы

1. Какие основные проблемы возникают при проектировании сложного ПО?
2. Какие уровни проектирования выделяют: система, подсистемы, классы, данные и методы?
3. Как абстракция, инкапсуляция и полиморфизм влияют на сложность программы?

4. Что означает слабая связность компонентов и почему она желательна?

5. Какие эвристические принципы и шаблоны проектирования применяются при конструировании?

6. Для чего используются UML-диаграммы и CRC-карточки?

7. Чем отличаются итеративный, нисходящий и восходящий подходы к проектированию?

Рубежные контрольные работы

Контрольная работа № 1. Основы конструирования ПО и подходы к разработке

Вариант выдаётся преподавателем. При сдаче представляются проектные артефакты и письменное обоснование решений.

1. На примере информационной системы описать проблему, границы проекта, связь требований с архитектурой и планом конструирования.

2. Сравнить две метафоры разработки применительно к выбранному проекту по критериям: цели, риски, изменения требований, сложность и возможности итераций.

3. Предложить и аргументировать комбинированный подход к конструированию; указать условия, при которых он предпочтительнее альтернатив.

Контрольная работа № 2. Предварительные условия, требования и архитектура

Вариант выдаётся преподавателем. При сдаче представляются проектные артефакты и письменное обоснование решений.

1. Определить тип заданной программной системы, выбрать последовательную либо итеративную модель и обосновать выбор.

2. Для описанной проблемной ситуации составить набор функциональных и нефункциональных требований, обработать два запроса на изменения по перечню критериев.

3. Построить высокоуровневую схему архитектуры, оценить состав предварительных работ и предложить измеримые метрики трудоёмкости.

Контрольная работа № 3. Выбор языков, технологий и методик конструирования

Вариант выдаётся преподавателем. При сдаче представляются проектные артефакты и письменное обоснование решений.

1. Сравнить минимум два языка программирования для заданного ПО по техническим, организационным и эксплуатационным критериям.

2. Составить короткую конвенцию кодирования, проверить фрагмент программы инструментом статического контроля, объяснить замечания.

3. Сравнить два фреймворка или архитектурных шаблона и выбрать технологию и процесс разработки, учитывая ограничения проекта.

Контрольная работа № 4. Проектирование компонентов и управление сложностью

Вариант выдаётся преподавателем. При сдаче представляются проектные артефакты и письменное обоснование решений.

1. Декомпозировать предложенную систему на подсистемы, классы и интерфейсы; нарисовать UML-диаграмму компонентов или классов.

2. Для ключевого класса подготовить CRC-карточку; указать ответственность, взаимодействия, точки изменений и потенциальную связность.

3. Предложить рефакторинг проблемного фрагмента кода с применением абстракции и инкапсуляции; объяснить, как уменьшается сложность.

Блок B. Задания реконструктивного уровня

- Модуль 1. Для заданного описания проекта найти несоответствия между поставленной проблемой, требованиями и выбранной метафорой разработки; предложить корректировку.
- Модуль 2. По исходному перечню требований провести их декомпозицию, обнаружить неоднозначности, уточнить запросы на изменение и исправить укрупнённую архитектуру.
- Модуль 3. По описанию технологического стека обнаружить неоправданные решения, предложить альтернативный язык/фреймворк и документировать конвенции кодирования.
- Модуль 4. По фрагменту диаграммы классов выявить избыточную связность и нарушения распределения ответственности, изменить проект с помощью UML и CRC.

Блок C. Практико-ориентированные задания

Практические работы составлены по 11 практическим занятиям РПД (по 2 академических часа каждое). При выполнении студент представляет отчёт, необходимые схемы или программные артефакты, результаты проверки и защищает выбранные решения.

Практическая работа № 1 (модуль 1). Подбор и комбинирование метафор конструирования. Для заданного проекта сравнить «написание кода», «выращивание системы» и «построение ПО»; составить матрицу критериев и предложить комбинированный подход.

Практическая работа № 2 (модуль 2). Определение типа ПО и модели разработки. Классифицировать заданное ПО, оценить требования к качеству и риски; обосновать последовательную или итеративную модель.

Практическая работа № 3 (модуль 2). Постановка проблемы и управление изменениями. Сформулировать проблему, видение, требования; проанализировать два запроса на изменения по контрольному списку.

Практическая работа № 4 (модуль 2). Высокоуровневая архитектура системы. Создать архитектурную схему с основными компонентами и интерфейсами; оценить полноту и риски решения.

Практическая работа № 5 (модуль 2). Оценка трудоёмкости и выбор метрик. Составить перечень предварительных работ, определить метрики времени и объёма, оценить трудоёмкость и ограничения оценки.

Практическая работа № 6 (модуль 3). Сравнение языков программирования. Выбрать язык для заданной разработки, сопоставив его с альтернативой по производительности, экосистеме, поддержке и рискам.

Практическая работа № 7 (модуль 3). Конвенции и проверка качества кода. Разработать правила стиля, выполнить проверку фрагмента кода статическим анализатором и исправить нарушения.

Практическая работа № 8 (модуль 3). Сравнение фреймворков и шаблонов. Сопоставить два фреймворка и два архитектурных шаблона применительно к одной задаче; обосновать выбор.

Практическая работа № 9 (модуль 3). Обоснование процесса и методик конструирования. Составить рекомендации по процессу разработки и набору методик с учётом размера команды, изменчивости требований и рисков.

Практическая работа № 10 (модуль 4). Итеративная проработка проектного решения. Выявить проблемы проектирования, показать две итерации улучшения решения и обосновать изменения.

Практическая работа № 11 (модуль 4). UML, CRC-карточки и документация компонентов. Построить UML-диаграммы, оформить CRC-карточки ключевых классов и кратко описать интерфейсы и ответственность.

Минимальные требования к отчёту по практической работе

- Цель, постановка задачи, исходные условия и ограничения.
- Выбранные критерии и инструменты, описание хода решения.
- Результат в проверяемой форме: таблица, требования, диаграмма, код, протокол проверки - по характеру задания.
- Анализ полученных результатов, выводы и обоснование альтернатив.
- Устная защита: демонстрация артефактов и ответы на вопросы.

Блок D. Промежуточная аттестация - зачёт с оценкой

Зачёт с оценкой проводится в 1-м семестре. Проверяются теоретическое понимание принципов конструирования, умение обосновывать технологический выбор и способность защищать проектное решение. Отдельная оценка зачёта - до 30 баллов.

Примерные зачётные билеты

Билет № 1. Теоретический вопрос: Содержание конструирования ПО, требования, архитектура. Практическая часть: Составить структуру плана конструирования небольшой системы.

Билет № 2. Теоретический вопрос: Метафоры разработки и факторы их выбора. Практическая часть: Обосновать комбинирование метафор для заданного проекта.

Билет № 3. Теоретический вопрос: Предварительные условия и типы ПО. Практическая часть: Выбрать модель разработки с учётом типа и ограничений системы.

Билет № 4. Теоретический вопрос: Требования, видение и запросы на изменения. Практическая часть: Проанализировать изменение требования и предложить решение.

Билет № 5. Теоретический вопрос: Метрики и оценка предварительных работ. Практическая часть: Составить набор метрик и пример оценки трудоёмкости.

Билет № 6. Теоретический вопрос: Языки, фреймворки и архитектурные шаблоны. Практическая часть: Обосновать технологический стек по предложенным критериям.

Билет № 7. Теоретический вопрос: Конвенции и контроль качества исходного кода. Практическая часть: Найти нарушения правил кодирования и предложить исправления.

Билет № 8. Теоретический вопрос: Принципы проектирования, слабая связность, UML и CRC. Практическая часть: Создать диаграмму компонентов и CRC-карточку одного класса.

Раздел 4. Методические материалы, процедуры и критерии оценивания

Оценивание проводится по фактически представленным и защищённым результатам. Максимальный рейтинг: 70 баллов за текущий и рубежный контроль и 30 баллов за зачёт с оценкой. Текущий контроль - практические задания; рубежный - индивидуальные контрольные работы.

Этап	Вид	Макс. балл	Показатели оценивания
Модуль 1	Текущий	6	Корректная постановка проектной задачи - 2; сравнение метафор - 2; аргументация решения - 2.
Модуль 1	Рубежный	6	Термины и подходы - 2; анализ альтернатив - 2; обоснование выбора - 2.
Модуль 2	Текущий	11	Тип ПО и модель процесса - 2; требования и изменения - 3; архитектура - 3; метрики и оценка трудоёмкости - 3.
Модуль 2	Рубежный	11	Постановка требований - 3; архитектурная схема - 4; обоснование и оценка - 4.
Модуль 3	Текущий	10	Обоснованный выбор языка - 2; стандарты кодирования - 3; выбор фреймворков - 3; методики разработки - 2.
Модуль 3	Рубежный	10	Анализ технологий - 4; критерии выбора - 3; аргументация применимости - 3.
Модуль 4	Текущий	8	Итеративное проектирование - 3; UML - 3; CRC и документация - 2.
Модуль 4	Рубежный	8	Декомпозиция и ответственность - 3; снижение сложности - 3; проектные выводы - 2.
Зачёт с оценкой	Итоговый	30	Теоретический вопрос - 10; индивидуальное проектное задание - 14; защита и ответы на вопросы - 6.

Пороговые значения и итоговая оценка

- За работу в семестре максимум 70 баллов; зачёт - максимум 30; общий рейтинг - 100 баллов.
- Зачётный минимум за семестр - 40 баллов, за зачёт - 20, суммарно - 60 баллов. Обязательные задания должны быть представлены и защищены.
- 85–100 баллов - «отлично»; 70–84 балла - «хорошо»; 60–69 - «удовлетворительно»; менее 60 - «неудовлетворительно».
- При оценке отдельной работы: 85–100% максимума - полное и аргументированное выполнение; 70–84% - несущественные недочёты; 60–69% - выполнены базовые требования, но допущены значимые недостатки; менее 60% - обязательный результат не подтверждён.
- Работы оцениваются по полноте исходных данных, качеству проектного артефакта, обоснованию альтернатив, корректности применённых методов и способности объяснить решения.

Процедура проведения контроля

Сроки и критерии доводятся преподавателем до начала модуля. Студент сдаёт отчёт и необходимые программные или проектные материалы, затем защищает их в ходе собеседования. Рубежная работа выполняется по индивидуальному варианту. Недочёты указываются преподавателем, а порядок повторной сдачи определяется локальными нормативными актами университета.

Раздел 5. Методические указания для обучающихся

5.1. Подготовка к текущему и рубежному контролю

- Для модуля 1 повторить понятия конструирования, требования, метафоры и методы разработки; подготовить сравнительную таблицу подходов.
- Для модуля 2 подготовить описание типа ПО, требования, решение по изменениям, архитектуру и обоснование трудоёмкости.
- Для модуля 3 представить матрицу выбора языка, анализ фреймворков, правила кодирования и проверяемые результаты контроля стиля.
- Для модуля 4 подготовить UML-диаграммы, CRC-карточки, пояснение распределения ответственности и способов снижения сложности.
- При выполнении самостоятельной работы использовать материалы лекций и литературу РПД, сохраняя ссылки на источники и фиксируя допущения.

5.2. Требования к представлению проектных материалов

Вся проектная документация должна позволять проследить цепочку «проблема - требования - архитектура - выбор технологий - проектное решение». Для UML следует указать тип диаграммы, назначение элементов и ключевые связи. При сравнении инструментов и подходов необходимо приводить критерии и ограничения применимости, а не только перечислять преимущества.

5.3. Подготовка к зачёту с оценкой

Рекомендуется подготовить портфолио выполненных практических заданий, одну целостную схему архитектуры, таблицу обоснования технологического стека, пример документации программного компонента и краткое пояснение реальных или предполагаемых рисков проекта. На защите необходимо аргументировать выбор и показать понимание альтернативных решений.

Контрольные работы и практические задания оцениваются в рамках текущего и рубежного контроля четырёх модулей.