

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ

Директор института прикладной  
математики и компьютерных наук

  
А.В. Замятин  
« 18 » \_\_\_\_\_ 2022 г.

Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине  
(Оценочные средства по дисциплине)

**Рефакторинг и обратное проектирование**

по направлению подготовки

**02.04.02 Фундаментальная информатика и информационные технологии**

Направленность (профиль) подготовки:

**Моделирование систем искусственного интеллекта**

ОС составил:  
ассистент кафедры программной инженерии



Л.С. Иванова

Рецензент: д-р физ.-мат. наук, профессор,  
профессор кафедры программной инженерии



О.А. Змеев

Оценочные средства одобрены на заседании учебно-методической комиссии  
института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от \_\_\_\_12.05\_\_\_\_ 2022 г. № \_\_4\_\_

Председатель УМК ИПМКН,  
д-р техн. наук, профессор



С.П. Сущенко

**Оценочные средства (ОС)** являются элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ОС разрабатываются в соответствии с рабочей программой (РП) дисциплины и включает в себя набор оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине.

### 1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

| Компетенция  | Индикатор компетенции  | Код и наименование результатов обучения (планируемые результаты обучения, характеризующие этапы формирования компетенций)                  | Критерии оценивания результатов обучения   |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  | Отлично  | Хорошо   | Удовлетворительно  | Неудовлетворительно  |
| ПК-1. Способен проектировать программное обеспечение | ИПК-1.1. Использует существующие типовые решения и шаблоны проектирования программного обеспечения | ОР-1.1.1. Знает основы объектно-ориентированного анализа и проектирования как современной методологии разработки программного обеспечения. | Знает основы объектно-ориентированного анализа и проектирования как современной методологии разработки программного обеспечения. | Имеет общее представление об основах объектно-ориентированного анализа и проектирования как современной методологии разработки программного обеспечения. | Имеет слабое представление об основах объектно-ориентированного анализа. | Не имеет представления об основах объектно-ориентированного анализа.             |
|  |  | ОР-1.1.2 Умеет находить типовые решения и шаблоны проектирования в артефактах проекта.   | Умеет четко находить типовые решения и шаблоны проектирования в артефактах проекта.  | Умеет находить типовые решения и шаблоны проектирования в артефактах проекта, но допускает незначительные ошибки.  | Допускает серьезные ошибки при поиске типовых решений.                   | Не умеет находить типовые решения и шаблоны проектирования в артефактах проекта. |

|  |  |   |   |   |   |   |
|--|--|---|---|---|---|---|
|  | ИПК-1.2. Применяет методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов | ОР-1.2.1. Владеет программными средствами проектирования и тестирования программного обеспечения.   | Владеет программными средствами проектирования и тестирования программного обеспечения на высоком уровне.   | Владеет не всеми требуемыми программными средствами проектирования.   | Владеет лишь некоторыми из требуемых программных средств проектирования.  | Не владеет программными средствами проектирования и тестирования программного обеспечения.  |
|  | ИПК-1.3. Знает методы и средства проектирования программного обеспечения, методы и средства проектирования баз данных              | ОР-1.3.1 Владеет методами проектирования программного обеспечения и баз данных.<br><br>ОР-1.3.2 Владеет методами обратного проектирования уже существующего программного обеспечения. | Владеет методами проектирования программного обеспечения и баз данных на высоком уровне.<br><br>Владеет методами обратного проектирования уже существующего программного обеспечения на высоком уровне. | Владеет методами проектирования программного обеспечения и баз данных, но допускает незначительные ошибки.<br><br>Владеет методами обратного проектирования уже существующего программного обеспечения, но допускает незначительные ошибки. | Допускает серьезные ошибки при использовании методов проектирования программного обеспечения и баз данных.<br><br>Допускает серьезные ошибки при использовании методов обратного проектирования уже существующего программного обеспечения. | Не владеет методами проектирования программного обеспечения и баз данных.<br><br>Не владеет методами обратного проектирования уже существующего программного обеспечения. |

|   |   |  |   |  |   |   |
|---|---|--|---|--|---|---|
| <p>ПК-2. Создает архитектурный проект программного средства</p> | <p>ИПК-2.1. Оценивает возможность создания архитектурного проекта программного средства</p> | <p>ОР-2.1.1 Знает методы оценки возможности создания архитектурного проекта программного средства.</p> <p>ОР-2.1.2 Владеет методами создания архитектурного проекта программного средства.</p> | <p>Знает всё, что требуется, о методах оценки возможности создания архитектурного проекта программного средства.</p> <p>Владеет методами создания архитектурного проекта программного средства на высоком уровне.</p> | <p>Владеет основной информацией о методах оценки возможности создания архитектурного проекта программного средства.</p> <p>Владеет методами создания архитектурного проекта программного средства, но допускает незначительные ошибки.</p> | <p>Допускает серьезные ошибки при применении знаний о методах оценки возможности создания архитектурного проекта программного средства.</p> <p>Допускает серьезные ошибки при применении методов создания архитектурного проекта программного средства.</p> | <p>Не знает о методах оценки возможности создания архитектурного проекта программного средства.</p> <p>Не владеет методами создания архитектурного проекта программного средства.</p> |
|---|---|--|---|--|---|---|

|  |  |   |   |   |  |  |
|--|--|---|---|---|--|--|
|  | ИПК-2.2. Определяет цели архитектуры программного средства | <p>ОР-2.2.1 Знает методы определения целей архитектуры программного средства.</p> <p>ОР-2.2.2 Владеет методами определения архитектурно-значимых целей программного средства.</p> | <p>Знает всё, что требуется о методах определения целей архитектуры программного средства.</p> <p>Владеет методами определения архитектурно-значимых целей программного средства на высоком уровне.</p> | <p>Владеет основной информацией о методах определения целей архитектуры программного средства.</p> <p>Владеет методами определения архитектурно-значимых целей программного средства, но допускает незначительные ошибки.</p> | <p>Допускает серьезные ошибки при применении знаний о методах определения целей архитектуры программного средства.</p> <p>Допускает серьезные ошибки при применении методов определения архитектурно-значимых целей программного средства.</p> | <p>Не знает о методах определения целей архитектуры программного средства.</p> <p>Не владеет методами определения архитектурно-значимых целей программного средства.</p> |
|--|--|---|---|---|--|--|

|  |  |   |  |  |  |   |
|--|--|---|--|--|--|---|
|  | <p>ИПК-2.3. Определяет ключевые сценарии для архитектуры программного средства</p> | <p>ОР-2.3.1 Владеет методами построения ключевых сценариев архитектуры программного средства.</p> <p>ОР-2.3.2 Владеет программными средствами для построения сценариев.</p> <p>ОР-2.3.3 Умеет строить архитектурные сценарии уже существующего программного средства.</p> | <p>Владеет методами построения ключевых сценариев архитектуры программного средства на высоком уровне.</p> <p>Владеет программными средствами для построения сценариев на высоком уровне.</p> <p>Умеет безошибочно строить архитектурные сценарии уже существующего программного средства.</p> | <p>Владеет методами построения ключевых сценариев архитектуры программного средства, но допускает незначительные ошибки.</p> <p>Владеет программными средствами для построения сценариев, но допускает незначительные ошибки.</p> <p>Умеет строить архитектурные сценарии уже существующего программного средства, но допускает незначительные ошибки.</p> | <p>Допускает серьезные ошибки при применении методов построения ключевых сценариев архитектуры программного средства.</p> <p>Допускает серьезные ошибки при применении программных средств для построения сценариев.</p> <p>Допускает серьезные ошибки при построении архитектурных сценариев уже существующего программного средства.</p> | <p>Не владеет методами построения ключевых сценариев архитектуры программного средства.</p> <p>Не владеет программными средствами для построения сценариев.</p> <p>Не умеет строить архитектурные сценарии уже существующего программного средства.</p> |
|--|--|---|--|--|--|---|

## Этапы формирования компетенций и виды оценочных средств

| №  | Этапы формирования компетенций (разделы дисциплины) | Код и наименование результатов обучения                              | Вид оценочного средства (тесты, задания, кейсы, вопросы и др.) |
|----|---|--|--|
| 1. | Качество программного обеспечения                   | ОР-1.1.1, ОР-1.1.2, ОР-1.3.1, ОР-1.3.2                               | Вопросы<br>Задания   |
| 2. | Тестирование  | ОР-1.2.1   | Вопросы<br>Задания   |
| 3. | Рефакторинг   | ОР-2.1.1, ОР-2.1.2, ОР-2.2.1, ОР-2.2.2, ОР-2.3.1, ОР-2.3.2, ОР-2.3.3 | Вопросы<br>Задания   |

### Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

Типовые задания для проведения текущего контроля успеваемости по дисциплине.

Для выполнения индивидуального проекта обучающимся предоставляется исходный код программной системы, заведомо содержащий проблемный код для его последующей идентификации и устранения.

Выполнение проекта ведется во время практических занятий и во время самостоятельной работы студента.

Текущий контроль осуществляется путем проверки для каждого студента выполнения задач по индивидуальному проекту. Также проводятся контрольные работы по теоретическому материалу по части вопросов из п. 3.2 в соответствии с пройденным материалом.

Типовые задания для проведения промежуточной аттестации по дисциплине.

Теоретические вопросы к зачету:

1. Что такое рефакторинг?
2. Какие проблемы можно решить с помощью рефакторинга?
3. Когда следует выполнять рефакторинг?
4. Как обосновать необходимость рефакторинга руководству?
5. Что такое code smell?
6. Дублированный код. Что такое, как его идентифицировать, какие проблемы он может вызывать?
7. Зачем нужно юнит-тестирование при рефакторинге.
8. Что необходимо покрывать юнит-тестами перед проведением рефакторинга.
9. Рефакторинг «Извлечение метода». Назначение. Способ применения. Пример.
10. Рефакторинг «Внедрение метода». Назначение. Способ применения. Пример.
11. Рефакторинг «Инкапсуляция коллекций». Назначение. Способ применения. Пример.
12. Рефакторинг «Замена магических чисел константами». Назначение. Способ применения. Пример.
13. Рефакторинг «Извлечение суперкласса». Назначение. Способ применения. Пример.
14. Рефакторинг «Извлечение интерфейса». Назначение. Способ применения. Пример.



15. Рефакторинг «Разделение иерархии наследования». Назначение. Способ применения. Пример.

16. Рефакторинг «Разделение интерфейса и бизнес-логики». Назначение. Способ применения. Пример.

### **Методические материалы, определяющие процедуры оценивания образовательных результатов обучения**

Методические материалы для оценки текущего контроля успеваемости по дисциплине.

Текущий контроль по лабораторным работам осуществляется в виде проверки выполнения заданий индивидуального проекта. Текущий контроль успеваемости по теоретическому материалу осуществляется в виде контрольных работ.

Оценка текущего контроля проводится на основе оценки компетенций, соответствующих текущему разделу дисциплины, согласно таблице раздела 1.

Методические материалы для проведения промежуточной аттестации по дисциплине.

Итоговая оценка по предмету (зачет с оценкой) выставляется следующим образом:

«отлично» – студент выполнил не менее 75% запланированных работ по индивидуальному проекту, нет неудовлетворительных оценок за контрольные работы, средняя (округленная) оценка за контрольные работы – «отлично»;

«хорошо» – студент выполнил не менее 75% запланированных работ по индивидуальному проекту, нет неудовлетворительных оценок за контрольные работы, средняя (округленная) оценка за контрольные работы – «хорошо»;

«удовлетворительно» – студент выполнил не менее 75% запланированных работ по индивидуальному проекту, нет неудовлетворительных оценок за контрольные работы, средняя (округленная) оценка за контрольные работы – «удовлетворительно»;

«неудовлетворительно» – студент не выполнил 75% запланированных работ по проекту или сдал хотя бы одну контрольную работу на «неудовлетворительно».

Во время зачета студент может повысить свою оценку, сдав заново соответствующую контрольную работу, при условии выполнения остальных требований к оценке.