

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Таскаев Сергей Валерьевич  
Должность: Ректор  
Дата подписания: 2025.12.16 09:00:59  
Уникальный программный ключ:  
054c0182970293149c21699f0009940292896684

МИНОБРАЗОВАНИЯ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Объектно-ориентированное программирование» по направлению подготовки (специальности) 02.03.02 «Фундаментальная информатика и информационные технологии» направленности (профилю) «Математические и алгоритмические основы интеллектуальных систем» ФГБОУ ВО «ЧелГУ»

стр. 1

**Фонд оценочных средств для промежуточной аттестации  
по дисциплине (модулю)**

**Объектно-ориентированное программирование**

Направление подготовки (специальность)

**02.03.02 Фундаментальная информатика и информационные технологии**

Направленность (профиль)

**Математические и алгоритмические основы интеллектуальных систем**

Присваиваемая квалификация (степень)

**бакалавр**

Форма обучения

**очная**

Год набора

**2026**

Троицк, 2026 г.



## Содержание

1. Паспорт фонда оценочных средств
2. Перечень формируемых компетенций
  - 2.1. Компетенции, закреплённые за дисциплиной
3. Содержание оценочных средств по дисциплине
  - 3.1. Виды оценочных средств
  - 3.2. Содержание оценочных средств
4. Порядок проведения и критерии оценивания промежуточной аттестации
  - 4.1. Порядок проведения промежуточной аттестации
  - 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств
  - 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций



## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки: *02.03.02 «Фундаментальная информатика и информационные технологии»*

Направленность (профиль): *Математические и алгоритмические основы интеллектуальных систем*

Дисциплина: *Объектно-ориентированное программирование*

Семестр изучения: *3*

Формы промежуточной аттестации: *зачет с оценкой*

Для оценивания результатов используется балльно-рейтинговая система

## 2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

### 2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Объектно-ориентированное программирование» направлено на формирование следующих компетенций:

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ПК-2	Способность к осуществлению интеграции программных модулей и компонент и проверки работоспособности программного продукта на основе международных и профессиональных стандартов информационных технологий, современных парадигм и методологий, инструментальных и вычислительных средств, методов и механизмов оценки и анализа функционирования средств и систем информационных	ПК-2.1. Обладает знаниями о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных, о создании программных интерфейсов; о методах и механизмах оценки и анализа функционирования средств и систем информационных технологий; о международных и профессиональных стандартах информационных	Знать: методы и средства объектно-ориентированного программирования (для достижения ПК-2.1) Уметь: применять методы и средства объектно-ориентированного программирования, создавать программные интерфейсы; проводить проверку и оценку работоспособности программного продукта (для достижения ПК-2.2) Владеть: навыками разработки процедур для развертывания программного обеспечения средствами объектно-ориентированного программирования (для достижения ПК-2.3)



	технологий	технологий, о современных парадигмах и методологиях, инструментальных и вычислительных средствах ПК-2.2. Демонстрирует умения: применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; проводить проверку и оценку работоспособности программного продукта. ПК-2.3. Имеет практический опыт (навыки): сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; оценки работоспособности программного продукта.	
ПК-3	Способность к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных	ПК-3.1. Обладает знаниями о методах и средствах проектирования программного обеспечения, структур данных, баз данных, программных	Знать: методы и средства проектирования программного обеспечения, программных интерфейсов (для достижения ПК-3.1) Уметь: разрабатывать требования к программному продукту, применять методы и



	технологий при решении проектно-технических и прикладных задач	интерфейсов. ПК-3.2. Демонстрирует умения: разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.3. Имеет практический опыт (навыки): проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	средства проектирования программного обеспечения, программных интерфейсов (для достижения ПК-3.2) Владеть: навыками проектирования программного обеспечения, структур данных, программных интерфейсов (для достижения ПК-3.3)
--	--	---	--

### 3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

#### 3.1 Виды оценочных средств

№ п/п	Контролируемые темы/разделы	Код компетенции/планируемые результаты обучения	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
1	Основные принципы ООП, объектно-ориентированная модель	ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, устный опрос	Зачет: Устный опрос (вопросы 1-2) Тестирование Практическое задание
2	Классы. Объекты. Методы. События	ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, устный опрос	Зачет: Устный опрос (вопросы 3-14, 21-25) Тестирование Практическое задание
3	Инструментальные средства ООП	ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, устный опрос	Зачет: Устный опрос (вопросы 15-20, 26-30) Тестирование Практическое



				задание
4	Разработка графического пользовательского интерфейса средствами ООП	ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, устный опрос	Зачет: Устный опрос (вопросы 31-33) Тестирование Практическое задание

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе по дисциплине. Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре и являются учебно-методическими материалами ограниченного (конфиденциального) пользования.

### 3.2 Содержание оценочных средств

Оценочные средства для промежуточной аттестации представлены базой контрольных вопросов, тестовых заданий и тематикой лабораторных работ.

#### 3.2.1. База контрольных вопросов к зачету

1. Базовые понятия объектно-ориентированного программирования.
2. Наследование, инкапсуляция, полиморфизм, понятие абстракции, классы.
3. Синтаксис объявления класса.
4. Использование класса.
5. Определение функции члена-класса.
6. Создание объекта класса.
7. Динамические объекты.
8. Неполное объявление класса. Спецификаторы доступа.
9. Конструкторы и деструкторы. Правила для конструкторов и деструкторов.
10. Список инициализации элементов.
11. Конструкторы копирования.
12. Встраиваемые функции.
13. Статические данные-члены класса. Константные данные-члены класса.
14. Статические функции-члены класса. Константные функции-члены класса.
15. Функции, дружественные одному классу.



16. Функции, дружественные нескольким классам.
17. Функции-члены, дружественные другому классу.
18. Простое наследование.
19. Множественное наследование.
20. Виртуальные базовые классы.
21. Перегрузка функций.
22. Перегрузка конструкторов.
23. Понятие и правила перегрузки операторов.
24. Перегрузка унарных операторов.
25. Перегрузка бинарных операторов.
26. Раннее и позднее связывание. Динамический полиморфизм.
27. Виртуальные функции.
28. Виртуальные деструкторы.
29. Абстрактные классы и чисто виртуальные функции.
30. Обработка исключительных ситуаций.
31. Графический пользовательский интерфейс.
32. Библиотеки для реализации графического интерфейса.
33. Графический интерфейс: событий, обработчик событий.

### 3.2.2. Тематика практических заданий зачета

1. Создание класса комплексных чисел и действий над ними.
2. Создание класса рациональных чисел и действий над ними.
3. Создание программы сортировки различного типа чисел и строк.
4. Создание функции обмена местами для переменных различных типов
5. Создание программ, работающих с линейными однонаправленными списками
6. Создание программ по перегрузке функций
7. Создание программ работающих с файлами и строковыми переменными
8. Конструктор / деструктор класса
9. Полиморфизм и переопределение операторов
10. Модули и их импорт
11. Приложения «Калькулятор», «Матричный калькулятор», «Комплексный калькулятор»
12. Приложения «Текстовый процессор», «Браузер»



## 13. Игровые приложения разных жанров

### 3.2.3. Типовое тестовое задание

1. Базовым понятием объектно-ориентированного программирования является:

a) метод; b) поле; c) ориентация; d) объект.

2. Наследование – это:

a) реализация только одной идеи устранения дублирования данных;

b) особое отношение между объектами одного класса;

c) такое отношение между классами, когда один класс повторяет структуру и поведение другого класса.

3. Механизм, который объединяет данные и методы, манипулирующие этими данными, и защищает и то и другое от внешнего вмешательства или неправильного использования, это:

a) наследование; b) полиморфизм; c) инкапсуляция; d) абстракция данных.

4. Класс, который получается в результате наследования, называется:

a) родительский; b) основной; c) порожденный.

5. С точки зрения объектно-ориентированной парадигмы программирования реальный мир состоит из ...

a) переменных b) классов c) объектов d) информации

6. Функция в языке программирования – это:

a) программный объект, задающий вычислительную процедуру определения значения от аргумента

b) сегмент программы, хранящий некоторое значение, зависящее от аргумента

c) программный объект, принимающий значение с помощью оператора присваивания Неверно

d) выражение, означающее зависимость левой части от правой



7. Базовым понятием объектно-ориентированного программирования является:

- a) метод b) поле c) объект d) ориентация

8. Основные принципы объектно-ориентированного программирования:

- a) Композиция b) Инкапсуляция c) Абстракция  
d) Иерархия e) Полиморфизм f) Наследование

9. Абстракция, в которой детали реализации некоторого действия скрываются за отдельным именем:

- a) процедура b) модуль c) метод d) функция

10. Наследование – это:

- a) такое отношение между классами, когда один класс повторяет структуру и поведение другого класса  
b) реализация только одной идеи устранения дублирования данных  
c) особое отношение между объектами одного класса

Ключ: 1-d; 2-c; 3-c; 4-c; 5-c; 6-a; 7-c; 9-b,c,e,f; 9-d; 10-a

## 4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### 4.1. Порядок проведения промежуточной аттестации

Промежуточная аттестация проводится в форме зачета в три этапа. На первом этапе студент выполняет компьютерный тест из 10 вопросов. Продолжительность – до 15 минут. На втором этапе студенту выдаётся теоретический вопрос по одному из разделов дисциплины из базы контрольных вопросов. Время выполнения – до 15 минут. На третьем этапе выдается практическое задание для выполнения на компьютере в рамках тематики одной из лабораторных работ. Время выполнения – до 40 минут.

При дистанционном обучении устный опрос, в том числе защита курсовых работ, проводятся в Microsoft Teams. Практические задания и письменные ответы размещаются в системе Moodle. Тестирование осуществляется в системе Moodle.



## 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

### 4.2.1. Критерии оценивания на зачете с оценкой

«Отлично» (91-100 баллов) – выставляется студенту в том случае, если он:

- глубоко и правильно усвоил программный материал, последовательно, грамотно и логически стройно его излагает;
- владеет основными методами и алгоритмами решения задач;
- умеет увязывать теорию с практикой, показывает умение применять знания;
- выполнил итоговый тест не менее чем на 75%.

«Хорошо» (81-90 баллов) – выставляется студенту, если он:

- твердо знает программный материал, грамотно и по существу его излагает;
- владеет основными методами;
- не допускает существенных ошибок, но испытывает затруднения в выводах;
- умеет применять основные положения и приемы для решения задач;
- выполнил итоговый тест не менее чем на 50%.

«Удовлетворительно» (65-80 баллов) – выставляется студенту в том случае, если он:

- имеет знания только основного материала, но не умеет делать выводов;
- допускает ошибки, недостаточно правильные формулировки;
- с трудом увязывает основные положения с практикой;
- выполнил итоговый тест не менее чем на 25%.

«Неудовлетворительно» (до 65 баллов) – выставляется студенту в том случае, если он:

- не знает основополагающих вопросов изучаемого курса или значительной части программного материала;
- допускает ошибки, обнаруживает неумение их исправлять;
- не может увязать теорию с практикой;
- выполнил итоговый тест менее чем на 25%.



### 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Код компетенции	Планируемые результаты обучения по дисциплине	Критерии оценивания			
		<b>Отлично</b> Высокий уровень освоения проверяемых компетенций	<b>Хорошо</b> Средний уровень освоения проверяемых компетенций	<b>Удовлетворительно</b> Базовый уровень освоения проверяемых компетенций	<b>Неудовлетворительно</b> Недостаточный уровень освоения проверяемых компетенций
ПК-2	<i>Знать:</i> методы и средства объектно-ориентированного программирования <i>Уметь:</i> применять методы и средства объектно-ориентированного программирования, создавать программные интерфейсы; проводить проверку и оценку работоспособности и программного продукта <i>Владеть:</i> навыками разработки процедур для развертывания программного обеспечения средствами объектно-ориентированного программирования	<i>Знает:</i> методы и средства объектно-ориентированного программирования <i>Умеет:</i> применять методы и средства объектно-ориентированного программирования, создавать программные интерфейсы; проводить проверку и оценку работоспособности и программного продукта <i>Владеет:</i> навыками разработки процедур для развертывания программного обеспечения средствами объектно-ориентированного программирования	<i>Знает:</i> методы и средства объектно-ориентированного программирования, но допускает несущественные ошибки <i>Умеет:</i> применять методы и средства объектно-ориентированного программирования, создавать программные интерфейсы; проводить проверку и оценку работоспособности и программного продукта, но допускает несущественные ошибки <i>Владеет:</i> навыками разработки процедур для развертывания программного обеспечения средствами объектно-ориентированного программирования, но допускает несущественные ошибки	<i>Знает:</i> в ограниченном объеме методы и средства объектно-ориентированного программирования <i>Умеет:</i> в ограниченном объеме применять методы и средства объектно-ориентированного программирования, создавать программные интерфейсы; проводить проверку и оценку работоспособности и программного продукта <i>Владеет:</i> в ограниченном объеме навыками разработки процедур для развертывания программного обеспечения средствами объектно-ориентированного программирования	<i>Не знает:</i> методы и средства объектно-ориентированного программирования <i>Не умеет:</i> применять методы и средства объектно-ориентированного программирования, создавать программные интерфейсы; проводить проверку и оценку работоспособности и программного продукта <i>Не владеет:</i> навыками разработки процедур для развертывания программного обеспечения средствами объектно-ориентированного программирования
ПК-3	<i>Знать:</i> методы и средства проектирования программного обеспечения, программных интерфейсов	<i>Знает:</i> методы и средства проектирования программного обеспечения, программных интерфейсов	<i>Знает:</i> методы и средства проектирования программного обеспечения, программных интерфейсов, но	<i>Знает:</i> в ограниченном объеме методы и средства проектирования программного обеспечения,	<i>Не знает:</i> методы и средства проектирования программного обеспечения, программных интерфейсов



	<i>Уметь:</i> разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, программных интерфейсов <i>Владеть:</i> навыками проектирования программного обеспечения, структур данных, программных интерфейсов	<i>Умеет:</i> разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, программных интерфейсов <i>Владеет:</i> навыками проектирования программного обеспечения, структур данных, программных интерфейсов	допускает несущественные ошибки <i>Умеет:</i> разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, программных интерфейсов, но допускает несущественные ошибки <i>Владеет:</i> навыками проектирования программного обеспечения, структур данных, программных интерфейсов, но допускает несущественные ошибки	программных интерфейсов <i>Умеет:</i> в ограниченном объеме разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, программных интерфейсов <i>Владеет:</i> в ограниченном объеме навыками проектирования программного обеспечения, структур данных, программных интерфейсов	<i>Не умеет:</i> разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения, программных интерфейсов <i>Не владеет:</i> навыками проектирования программного обеспечения, структур данных, программных интерфейсов
--	--	--	---	---	--

Уровни сформированности компетенций определяются следующим образом:

1. Высокий уровень соответствует оценке “отлично” (“зачтено”), и предполагает:

- готовность к самостоятельной профессиональной деятельности;
- глубокое и правильное усвоение программного материала, последовательное, грамотное и логически стройное его изложение;
- владение основными методами и алгоритмами решения задач;
- умение строить математические модели, увязывать теорию с практикой, применять знания.

2. Средний уровень соответствует оценке “хорошо” (“зачтено”) и предполагает:

- твердое знание программного материала, его изложение грамотное и по существу;
- владение основными методами;
- отсутствие существенных ошибок, но затруднения в выводах и доказательствах;
- умение применять основные положения для решения задач.



3. Базовый уровень соответствует оценке “удовлетворительно” (“зачтено”), и предполагает:

- знания только основного материала, неумение делать выводы и проводить доказательства;
- ошибки, недостаточно правильные формулировки;
- трудное увязывание основных положений с практикой.

4. Низкий уровень соответствует оценке “неудовлетворительно” (“не зачтено”) и предполагает:

- незнание основополагающих вопросов изучаемого курса или значительной части программного материала;
- ошибки, неумение их исправлять;
- неумение увязать теорию с практикой.

