

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Т.Ф.ГОРБАЧЕВА»  
Филиал КузГТУ в г. Белово



УТВЕРЖДАЮ

Директор филиала

*И.К. Костин*

И.К. Костин

31 » 08 20 21 г.

**Фонд оценочных средств по дисциплине**

**Объектно-ориентированное программирование и разработка информационных систем**

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль 01 «Прикладная информатика в экономике»

Присваиваемая квалификация "Бакалавр"

Белово 2021

ФОС составил ст. преподаватель Е. Аксент Е.Г. Аксененко


ФОС обсужден на заседании кафедры горного дела и техносферной безопасности

Протокол № 10 от « 15 » 06 2021 г.

Зав. кафедрой горного дела и техносферной безопасности  В.Ф. Белов

Согласовано учебно-методическим советом филиала КузГТУ в г. Белово

Протокол № 11 от « 22 » 06 2021 г.

Председатель учебно-методического совета  Ж.А. Долганова



# **1 Перечень планируемых результатов обучения по дисциплине "Объектно-ориентированное программирование и разработка информационных систем", соотнесенных с планируемыми результатами освоения образовательной программы**

Освоение дисциплины направлено на формирование:

профессиональных компетенций:

ПК-1 - Способность проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе.

ПК-10 - Способность принимать участие в организации ИТ-инфраструктуры и управлении информационной безопасностью.

ПК-2 - Способность разрабатывать и адаптировать прикладное программное обеспечение.

ПК-6 - Способность принимать участие во внедрении информационных систем.

ПК-7 - Способность настраивать, эксплуатировать и сопровождать информационные системы и сервисы.

ПК-8 - Способность проводить тестирование компонентов программного обеспечения ИС.

ПК-9 - Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

**Результаты обучения по дисциплине определяются индикаторами достижения компетенций**

**Индикатор(ы) достижения:**

Проводит обследование организаций, выявляет информационные потребности пользователей, формирует требования к информационной системе.

Разрабатывает и адаптирует прикладное программное обеспечение.

Принимает участие во внедрении информационных систем.

Настраивает, эксплуатирует и сопровождает информационные системы и сервисы.

Проводит тестирование компонентов программного обеспечения ИС.

Осуществляет ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

Принимает участие в организации ИТ-инфраструктуры и управлении информационной безопасностью.

**Результаты обучения по дисциплине:**

Знает:

- методологии, модели и технологии проектирования информационных систем; проектирование обеспечивающих подсистем ИС; методы обследования организаций; способы формализованного описания систем; методы спецификации требований к информационной системе;

- принципы организации проектирования и содержание этапов процесса разработки программных комплексов

- задачи и методы исследования и обеспечения качества и надежности программных компонентов; экономико-правовые основы разработки программных продуктов;

- принципы организации проектирования и содержание этапов процесса разработки программных комплексов;

- модели данных; архитектуру БД; системы управления БД и информационными хранилищами; методы и средства проектирования БД, особенности администрирования БД в локальных и глобальных сетях;

- принципы организации проектирования ИС; содержание этапов процесса разработки, внедрения, адаптации и настройки программных комплексов.

Умеет:

- использовать методы обследования организаций для выявления информационных потребностей пользователей; выполнять формализованное описание предметной области; формировать требования к информационной системе; документировать требования к информационной системе.

- разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования

- выбирать методы моделирования систем, структурировать и анализировать цели и функции систем управления, проводить системный анализ прикладной области

- разрабатывать концептуальную модель прикладной области, выбирать инструментальные средства и технологии проектирования ИС; проводить формализацию и реализацию решения прикладных задач;

- формулировать требования к создаваемым программным комплексам;

- выполнять работы на всех стадиях жизненного цикла проекта ИС, оценивать качество и затраты проекта;

- внедрять, адаптировать и настраивать ИС.

Владеет:

- навыками построения объектно-ориентированных моделей предметной области; навыками документирования требований к информационной системе.

- навыками работы в современной программно-технической среде в различных операционных системах; разработки программных комплексов для решения прикладных задач, оценки сложности алгоритмов и программ, использования современных технологий программирования, тестирования и документирования программных комплексов работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов;

- навыками работы с инструментальными средствами проектирования баз данных и знаний, управления проектами ИС и защиты информации;

навыками программирования и администрирования ИС.

## 2. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине " Объектно-ориентированное программирование и разработка информационных систем"

### 2.1. Паспорт фонда оценочных средств

Форма(ы) текущего контроля	Компетенции, формируемые в результате освоения дисциплины	Индикатор (ы) достижения компетенции	Результаты обучения по дисциплине (модулю)	Уровень достижения компетенции
Подготовка отчетов по практическим и (или) лабораторным работам, защита работ, тестирование	ПК-1 ПК-2 ПК-6 ПК-7 ПК-8 ПК-9 ПК-10	Проводит обследование организаций, выявляет информационные потребности пользователей, формирует требования к информационной системе. Разрабатывает и адаптирует прикладное программное	Знает: - методологии, модели и технологии проектирования информационных систем; проектирование обеспечивающих подсистем ИС; методы обследования организаций; способы формализованного описания систем; методы спецификации требований к информационной системе; - принципы организации проектирования и содержание этапов процесса разработки программных комплексов - задачи и методы исследования	Высокий или средний

		<p>обеспечение. Принимает участие во внедрении информационных систем. Настраивает, эксплуатирует и сопровождает информационные системы и сервисы. Проводит тестирование компонентов программного обеспечения ИС. Осуществляет ведение баз данных и поддержку информационного обеспечения решения прикладных задач. Принимает участие в организации ИТ-инфраструктуры и управлении информационной безопасностью.</p>	<p>и обеспечения качества и надежности программных компонентов; экономико-правовые основы разработки программных продуктов; - принципы организации проектирования и содержание этапов процесса разработки программных комплексов; - модели данных; архитектуру БД; системы управления БД и информационными хранилищами; методы и средства проектирования БД, особенности администрирования БД в локальных и глобальных сетях; - принципы организации проектирования ИС; содержание этапов процесса разработки, внедрения, адаптации и настройки программных комплексов. Умеет: - использовать методы обследования организаций для выявления информационных потребностей пользователей; выполнять формализованное описание предметной области; формировать требования к информационной системе; документировать требования к информационной системе. - разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования - выбирать методы моделирования систем, структурировать и анализировать цели и функции систем управления, проводить системный анализ прикладной области - разрабатывать концептуальную модель прикладной области, выбирать инструментальные средства и технологии проектирования ИС; проводить формализацию и реализацию решения прикладных задач; - формулировать требования к создаваемым программным комплексам; - выполнять работы на всех стадиях жизненного цикла проекта ИС, оценивать качество и затраты проекта; - внедрять, адаптировать и настраивать ИС. Владеет: - навыками построения</p>	
--	--	---	---	--

			<p>объектно-ориентированных моделей предметной области; навыками документирования требований к информационной системе.</p> <ul style="list-style-type: none"> <li>- навыками работы в современной программно-технической среде в различных операционных системах; разработки программных комплексов для решения прикладных задач, оценки сложности алгоритмов и программ, использования современных технологий программирования, тестирования и документирования программных комплексов работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов;</li> <li>- навыками работы с инструментальными средствами проектирования баз данных и знаний, управления проектами ИС и защиты информации;</li> <li>- навыками программирования и администрирования ИС.</li> </ul>	
<p><b>Высокий уровень достижения компетенции</b> - компетенция сформирована частично, рекомендованные оценки: отлично, хорошо, зачтено.</p> <p><b>Средний уровень достижения компетенции</b> - компетенция сформирована частично, рекомендованные оценки: хорошо, удовлетворительно, зачтено.</p> <p><b>Низкий уровень достижения компетенции</b> - компетенция не сформирована частично, оценивается неудовлетворительно или не зачтено.</p>				

## 2.2. Типовые контрольные задания или иные материалы

Текущий контроль успеваемости и промежуточная аттестация обучающихся могут проводиться как при непосредственном взаимодействии педагогического работника с обучающимися, так и с использованием ресурсов ЭИОС филиала КузГТУ, в том числе синхронного и (или) асинхронного взаимодействия посредством сети «Интернет».

## 2.3. Оценочные средства при текущем контроле

Для целей текущей аттестации используется выдача индивидуального дополнительного задания, обеспечивающего приведения кода к корректному виду, соответствующему объектную декомпозицию. Формулировка задания осуществляется непосредственно при анализе полученного кода исполнения лабораторной работы.

Текущий контроль по дисциплине заключается также в оформлении и защите отчетов по лабораторным работам, тестировании.

### Вопросы на защиту лабораторных работ.

1. Перечислите классические типы. Опишите механизмы для создания новых типов.
2. В чем различие между видами и методами (способами) абстракции?
3. Дайте характеристику парадигме ООП и специфике интерфейса ОО-программ.
4. Дайте понятие класса в ООП. Опишите отношение "объект - класс".
5. Опишите механизм наследования.

6. Дайте характеристику ООП в 4GL, его отличия от 3GL и 2GL
7. Дайте характеристику ООП в 2GL (байт-кодах), его отличия от 3GL и 4GL
8. Что такое дружественная функция?
9. Опишите роль ООП и объектно-событийной парадигмы в сетевом программировании.
10. Дайте характеристику ООП в 3GL, его отличия от 4GL и 2GL
11. Что такое аргумент по умолчанию? Как аргумент по умолчанию связан с перегрузкой функций?
12. Когда следует в программе применять шаблоны, а когда нет?
13. Для каких типов данных может применяться конкретный шаблон, а для каких нет?
14. Дайте определение и опишите способы и место применения технологии внедрения объектов с помощью макросов VBA.
15. Опишите организацию файлового и строкового потока.
16. В чем заключается отличия технологии объектно-ориентированных СУБД от реляционных?
17. Опишите роль ООП, объектно-событийной парадигмы, объекта в процессах параллельных вычислений.
18. Объясните, чем отличается восходящий парсинг от нисходящего?
19. Опишите принцип работы ООП в экспертных системах.
20. Опишите принцип работы объектно-ориентированных СУБД, их отличие от реляционных.
21. В чем смысл декомпозиции при составлении программ?
22. Опишите преимущества ООП в сравнении с классическим программированием.
23. Сформулируйте понятие полиморфизма. Проявления полиморфизма.
24. Что такое событие? Приведите пример на JavaScript.
25. Дайте определение и опишите способы и место применения технологии связи и внедрения объектов OLE v1.0 и OLE v2.0.
26. В чем разница между public, private, protected?
27. Дайте определение и опишите способы и место применения технологии связи и внедрения объектов ActiveX (OLE v3.0)
28. Что такое виртуальная функция?
29. Что такое абстрактный класс и чем может быть вызвана необходимость построений абстрактного класса?
30. Что такое раннее и позднее связывание?
31. Чем шаблоны лучше макроподстановок?
32. В чем основное преимущество использования библиотеки потокового ввода-вывода?
33. Дайте понятие объекта в ООП. Опишите отношение "объект - класс".
34. Как изменяются атрибуты элементов класса при наследовании?
35. Перечислите известные Вам платформы виртуальных машин и дайте характеристику их технологий.
36. В чем заключается смысл инкапсуляции?
37. Что такое конструктор, деструктор и когда они вызываются?
38. Дайте общее понятие интерфейса, виды интерфейсов и понятие класса интерфейсов в ООП.
39. Опишите технологию применения ООП в клиент-серверных системах.
40. Опишите принципы классификации языков ООП по областям применения. Приведите примеры.
41. Опишите назначение, свойства, недостатки и преимущества ОО-языка Visual Basic. Приведите пример кода.
42. Опишите назначение, свойства, недостатки и преимущества ООП в языке PHP. Приведите пример кода.
43. Опишите назначение, свойства, недостатки и преимущества ОО-языка Java. Приведите пример кода.



44. Опишите назначение, свойства, недостатки и преимущества ОО-языка JavaScript. Приведите пример кода.
45. Опишите назначение, свойства, недостатки и преимущества ОО-языка ActionScript. Приведите пример кода.
46. Опишите назначение, свойства, недостатки и преимущества ОО-языка С-шарп. Приведите пример кода.
47. Опишите назначение, свойства, недостатки и преимущества ОО-языка Object Pascal / Delphi. Приведите пример кода.
48. Опишите назначение, свойства, недостатки и преимущества ОО-языка Python. Приведите пример кода
49. Опишите назначение, свойства, недостатки и преимущества ОО-языка XML. Приведите пример кода.
50. Опишите назначение, свойства, недостатки и преимущества ОО-языка С++. Приведите пример кода.
51. Опишите назначение, свойства, недостатки и преимущества ОО-языка SQL3. Приведите пример кода.
52. Опишите назначение, свойства, недостатки и преимущества ОО-языка АВАР/4. Приведите пример кода.
53. Опишите назначение, свойства, недостатки и преимущества ОО-языка Objective-C. Приведите пример кода.
54. Опишите назначение, свойства, недостатки и преимущества ОО-языка 1С. Приведите пример кода.
55. Опишите назначение, свойства, недостатки и преимущества ОО-языка Ada. Приведите пример кода.
56. Опишите назначение, свойства, недостатки и преимущества ОО-языка Дракон. Приведите пример кода.
57. Опишите назначение, свойства, недостатки и преимущества ОО-языка VBA. Приведите пример кода.
58. Опишите назначение, свойства, недостатки и преимущества ОО-языка VBScript. Приведите пример кода.
59. Опишите назначение, свойства, недостатки и преимущества ОО-языка Visual Basic .Net. Приведите пример кода.
60. Опишите назначение, свойства, недостатки и преимущества ОО-модели (DOM) языка HTML. Приведите пример кода.

Критерии оценивания:

Критерии оценивания

- 90–100 баллов – при правильном выполнении заданий лабораторной работы, правильном ответе на все заданные вопросы;
- 80–89 баллов – при правильном выполнении заданий лабораторной работы, недостаточно полных ответов на заданные вопросы;
- 60–79 баллов – при неполном выполнении заданий лабораторной работы и/или неправильных, неточных ответах на вопросы;
- 0–59 баллов – при наличии серьезных ошибок при выполнении заданий лабораторной работы, неправильных ответах на вопросы или отсутствии выполненного задания и/или ответов на вопросы.

Шкала оценивания

Количество баллов	0–59	60–79	80-89	90-100
Шкала оценивания	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично

**Тестирование****Пример тестовых заданий для проведения текущего контроля**

1. Задачи какого рода решают основные шаблоны проектирования? ·

Повседневные задачи: переадресация работы, обеспечение слабой связности системы и так далее ·

Абстрагирование процесса создания экземпляров класса ·

Создание различных структур, изменяющих поведение уже существующих объектов ·

Определяют алгоритмы и способы взаимодействия объектов между собой

2. Задачи какого рода решают поведенческие шаблоны проектирования? ·

Повседневные задачи: переадресация работы, обеспечение слабой связности системы и так далее ·

Создание различных структур, изменяющих поведение уже существующих объектов ·

Абстрагирование процесса создания экземпляров класса ·

Определяют алгоритмы и способы взаимодействия объектов между собой

3. Задачи какого рода решают порождающие шаблоны проектирования? ·

Повседневные задачи: переадресация работы, обеспечение слабой связности системы и так далее ·

Абстрагирование процесса создания экземпляров класса ·

Создание различных структур, изменяющих поведение уже существующих объектов ·

Определяют алгоритмы и способы взаимодействия объектов между собой

4. Задачи какого рода решают структурные шаблоны проектирования? ·

Повседневные задачи: переадресация работы, обеспечение слабой связности системы и так далее ·

Абстрагирование процесса создания экземпляров класса ·

Создание различных структур, изменяющих поведение уже существующих объектов ·

Определяют алгоритмы и способы взаимодействия объектов между собой

5. Каким символом на диаграмме классов изображается тот факт, что атрибут имеет область видимости private?

ответ \_\_\_\_\_

6. Каким символом на диаграмме классов изображается тот факт, что атрибут имеет область видимости protected?

ответ \_\_\_\_\_

7. Каким символом на диаграмме классов изображается тот факт, что атрибут имеет бесконечную кратность?

ответ \_\_\_\_\_

8. Каким символом на диаграмме классов изображается тот факт, что атрибут имеет область видимости public?

ответ \_\_\_\_\_

9. Верно ли то, что отношение композиции является частным случаем отношения агрегации? ·

Да · Нет

10. Верно ли то, что существуют такие классы, которые могут иметь один и только один экземпляр? ·

Да · Нет

11. Верно ли, что понятие «суперкласс» является синонимом родительского класса? ·

Да · Нет 1

12. Верно ли, что понятия «вложенный класс» и «класс-член» являются синонимами? ·

Да · Нет

13. Верно ли, что программист не может управлять сборкой мусора? ·

Да · Нет 1

4. Верно ли, что концепция MVC включает в себя ровно три структурных составляющих: модель, вид и контроллер? ·

Да · Нет

15. Верно ли, что наиболее общим отношением между классами является отношение зависимости? · Да · Нет

16. Верно ли утверждение о том, что Java является декларативным языком программирования? ·

Да · Нет

17. Верно ли утверждение о том, что Java является императивным языком программирования? ·

Да · Нет

18. Верно ли, что Java позволяет создавать массив заранее известных величин? ·

Да · Нет

19. Верно ли, что MVC позволяет визуализировать одни и те же данные разными способами без изменения модели? ·

Да · Нет

20. Верно ли, что MVC позволяет подключать к одному виду несколько контроллеров? ·

Да · Нет

21. Верно ли, что для создания дескруктора в Java используется блок finalize? ·

Да · Нет

22. Верно ли, что один абстрактный класс не может являться родительским по отношению к другому абстрактному классу? ·

Да · Нет

23. Верно ли, что абстрактный класс не может иметь экземпляров? ·

Да · Нет

24. Верно ли, что абстрактный класс не может содержать реализаций методов? ·

Да · Нет

25. Верно ли, что в активной модели MVC об изменениях данных представление оповещает контроллер? ·

Да · Нет

26. Верно ли, что в пассивной модели MVC об изменениях данных представление оповещает контроллер? ·

Да · Нет

27. Верно ли, что основная цель использование концепции MVC – разграничение логики работы и визуализации? ·

Да · Нет

28. Верно ли, что основная цель использование концепции MVC – упрощение архитектуру приложения? ·

Да · Нет

29. Верно ли, что отношение агрегации может применяться для изображения иерархических отношений между классами? ·

Да · Нет

30. Как расшифровывается аббревиатура JRE?

ответ \_\_\_\_\_

31. Как расшифровывается аббревиатура JDK?

ответ \_\_\_\_\_

32. Как расшифровывается аббревиатура JVM?

33. Какая из парадигм программирования является ведущей в языке Java? ·

Объектно-ориентированная ·

Процедурная ·

Событийно-ориентированная ·

Компонентно-ориентированная

34. Какая команда используется для выполнения байт-кода? ·

java ·  
javac ·  
javar ·  
javargun

35. Какие группы шаблонов проектирования обычно выделяют? ·

Основные ·  
Порождающие ·  
Структурные ·  
Поведенческие

36. Какие задачи позволяет решать ключевое слово final? ·

Создание ненаследуемых методов ·  
Создание абстрактных классов ·  
Создание непереопределяемых методов ·  
Создание неизменяемых полей ·  
Создание классов, которые не могут иметь наследников

37. Какие из перечисленных требований должны выполняться для создания шаблона «неизменяемый объект» в Java? ·

Все поля класса должны иметь модификатор private ·  
Все методы класса должны быть объявлены с модификатором static ·  
Ссылка this не должна передаваться вовне во время конструирования объекта ·  
Класс должен быть объявлен с модификатором final

38. Какие из перечисленных утверждений касательно высокоуровневых шаблонов проектирования верны? ·

Высокоуровневые шаблоны также называют архитектурными ·  
Высокоуровневые шаблоны являются универсальными решениями ·  
Несмотря на то, что высокоуровневые шаблоны являются универсальными, они учитывают специфические особенности конкретных языков программирования ·

Высокоуровневые шаблоны также называют идиомами

39. Какие из перечисленных утверждений касательно низкоуровневых шаблонов проектирования верны? ·

Низкоуровневые шаблоны учитывают специфику языка программирования ·  
Низкоуровневые шаблоны также называют идиомами ·  
Низкоуровневые шаблоны являются универсальными ·  
Низкоуровневые шаблоны не используются в прикладных разработках

40. Какие из предлагаемых утверждений верны? ·

Статическим называется член класса, доступ к которому возможен без создания экземпляра ·  
Создание статического метода возможно только при использовании ключевого слова static ·  
Статические поля являются общими для всех экземпляров класса ·  
Статические поля недоступны экземплярам класса

41. Какие типы связей используют в UML? ·

Зависимость ·  
Ассоциация ·  
Обобщение ·  
Реализация

42. Какие типы сущностей выделяют в UML? ·

Поведенческие ·  
Аннотирующие ·  
Структурные ·  
Группирующие

43. Каково основное назначение статического импорта? ·

Позволяет JVM эффективно перенаправлять запросы к классам из других пакетов, принимая во внимание их статическую структуру ·

Позволяет обращаться к статическим членам без явного указания класса-владельца ·

Добавляет в область видимости описываемого класса все статические структуры из указанного пакета

44. Какое из перечисленных утверждений верно? ·

UML создан путем объединения всех существующих подходов проектирования того времени ·

UML создан на основе наиболее популярных подходов проектирования того времени ·

UML является исторически первым языком моделирования

45. Какое ключевое слово может быть использовано для обращения экземпляра класса к самому себе? ·

this ·

super ·

instanceof ·

own

46. Какое количество секций может содержать блок описания интерфейса на UML-диаграмме классов? ·

Одну ·

Две ·

Три ·

Четыре

47. Какое количество секций может содержать блок описания класса на UML-диаграмме классов? ·

Одну ·

Три ·

Две ·

Четыре

48. Как расшифровывается аббревиатура UML? ·

Unified Modeling Language ·

Universally Modeling Language ·

Unified Modern Language ·

Universally Modern Luggage

49. Что из перечисленного может быть отнесено к недостаткам шаблона «отложенная инициализация»? ·

Не позволяет явным образом задать порядок инициализации ·

Вносит дополнительную задержку при обращении к объекту ·

Порождает дополнительные ошибки при многопоточной работе ·

Замедляет инициализацию объектов

50. Что из перечисленного может быть отнесено к преимуществам шаблона «адаптер»? ·

Увеличивает степень независимости системы от сторонних библиотек ·

Смена библиотек не требует изменения всей системы ·

Ускорение начальной инициализации ·

Минимизация задержки при обращении

51. Что из перечисленного может быть отнесено к преимуществам шаблона «делегирование»? ·

Позволяет изменить поведение избегая наследования ·

Повышает степень абстракции ·

Положительно сказывается на времени обработки запроса системой ·

Сокращает число элементов в адресном пространстве

52. Что из перечисленного может быть отнесено к преимуществам шаблона «декоратор»? ·

Позволяет динамически добавлять объектам новые обязанности ·

Упрощает структуру отдельных классов ·

Сокращает количество имен в адресном пространстве ·

Не требует дополнительной иерархии классов

53. Что из перечисленного может быть отнесено к преимуществам шаблона «заместитель»? ·

Позволяет инициализировать объекты по требованию ·  
Позволяет придать объектам дополнительный функционал ·  
Сокращает время отклика системы ·  
Не требует создания новых объектов

54. Что из перечисленного может быть отнесено к преимуществам шаблона «итератор»? ·

Позволяет поддерживать единообразный интерфейс для доступа к элементам системы ·

Позволяет поддерживать несколько активных обходов коллекции ·

Итератор не должен владеть всей информацией о коллекции ·

Не требует дополнительной иерархии классов

55. Что из перечисленного может быть отнесено к преимуществам шаблона «команда»? ·

Позволяет параметризовать объекты выполняемым действием ·

Позволяет обрабатывать запросы как объекты ·

Упрощает иерархию классов ·

Уменьшает время отклика системы

56. Что из перечисленного может быть отнесено к преимуществам шаблона «компоновщик»? ·

Упрощает архитектуру приложения ·

Упрощает процедуру добавления новых компонентов в систему ·

Позволяет единообразно работать с составными и примитивными структурами ·

Позволяет ограничить типы используемых компонентов

57. Что из перечисленного может быть отнесено к членам класса? ·

Вложенный класс ·

Метод ·

Объект ·

Поле

58. Что из перечисленного может быть отнесено к преимуществам шаблона «наблюдатель»? ·

Сокращает цепочку вызовов, необходимую для уведомления наблюдателей об изменении состояния объекта ·

Абстрагирует связность объекта и наблюдателей ·

Позволяет обеспечить широковебательную рассылку уведомлений ·

Повышает степень повторной используемости кода

59. Что из перечисленного может быть отнесено к преимуществам шаблона «неизменяемый объект»? ·

Позволяет предотвратить ряд ошибок ·

Может быть использован в качестве ключа коллекции ·

Удобен для многопоточной работы ·

Положительно сказывается на скорости работы

60. Что из перечисленного может быть отнесено к преимуществам шаблона «хранитель»? ·

Позволяет сохранять внутренне состояние объекта ·

Упрощает структуру класса-владельца ·

Не раскрывает детали реализации класса-владельца ·

Позволяет восстанавливать состояние класса-владельца

61. Что из перечисленного может быть отнесено к преимуществам шаблона «цепочка ответственности»? ·

Гарантирует, что рано или поздно запрос будет обработан ·

Позволяет ослабить связи внутри системы ·

Не требует дополнительной иерархии классов ·

Придает дополнительную гибкость при распределении обязанностей

62. Что из перечисленного может быть отнесено к преимуществам шаблона «одиночка»? ·

Сокращение числа имен в глобальном адресном пространстве ·

Предоставление глобальной точки доступа ·

Ускорение начальной инициализации ·

Минимизация задержки при обращении в силу статической сущности

63. Что из перечисленного может быть отнесено к преимуществам шаблона «посредник»? ·  
 Централизует управление ·  
 Обеспечивает слабую связность системы ·  
 Упрощает протоколы взаимодействия между компонентами ·  
 Повышает степень повторной используемости кода
64. Что из перечисленного может быть отнесено к преимуществам шаблона «прототип»? ·  
 Позволяет специфицировать новые объекты путем изменения значений ·  
 Позволяет сократить иерархию классов ·  
 Позволяет динамически конфигурировать приложение классами ·  
 Значительно уменьшает время, необходимое на создание новых объектов
65. Что из перечисленного может быть отнесено к преимуществам шаблона «состояние»? ·  
 Изолирует логику работы от реализации ·  
 Не требует дополнительных имен в адресном пространстве ·  
 Позволяет полностью скрыть информацию о классе-владельце ·  
 Позволяет избегать применения цепочек условных операторов
66. Что из перечисленного может быть отнесено к преимуществам шаблона «стратегия»? ·  
 Позволяет оперировать семейством алгоритмов ·  
 Предотвращает порождение большого числа подклассов ·  
 Сокращает цепочки условных операторов, реализующих сложное поведение системы ·  
 Позволяет динамически выбирать реализацию алгоритма
67. Что из перечисленного может быть отнесено к преимуществам шаблона «строитель»? ·  
 В сравнении с фабричным методом придает дополнительную гибкость процессу конструирования объектов ·  
 Изолирует код конструирования объектов ·  
 Позволяет изменять внутреннее представление объекта ·  
 Не требует дополнительной иерархии классов
68. Что из перечисленного может быть отнесено к преимуществам шаблона «фабричный метод»? ·  
 Положительно сказывается на скорости работы ·  
 Позволяет устанавливать связь между параллельными иерархиями классов ·  
 Сокращает количество имен в адресном пространстве ·  
 Позволяет сделать код создания объектов более универсальным
69. Что из перечисленного может быть отнесено к преимуществам шаблона «фасад»? ·  
 Изолирует клиентов от подсистем ·  
 В большинстве случаев приводит к сокращению числа имен в адресном пространстве клиентов  
 Облегчает устройство системы ·  
 Ограничивает доступ клиентов к подсистемам
70. Каким образом будет выглядеть команда для компиляции класса test, описанного в файле test.java?  
 ответ \_\_\_\_\_
71. Сколько байт в памяти ЭВМ занимает одна переменная типа char?  
 ответ \_\_\_\_\_
72. Как называется один из фундаментальных механизмов объектно-ориентированного программирования, позволяющий классу ограничивать доступ к своим членам?  
 ответ \_\_\_\_\_
73. Как называется один из фундаментальных механизмов объектно-ориентированного программирования, позволяющий одному классу расширять функциональность другого, заимствуя при этом поля и методы?  
 ответ \_\_\_\_\_

74. Как называется один из фундаментальных механизмов объектно-ориентированного программирования, позволяющий элементам с одинаковой спецификацией иметь различную реализацию?

ответ \_\_\_\_\_

75. Как называется специализированный метод класса, предназначенный для создания нового экземпляра?

ответ \_\_\_\_\_

76. Как называется специализированный метод класса, предназначенный для уничтожения существующего экземпляра?

ответ \_\_\_\_\_

77. Какое ключевое слово используется для указания класса-родителя?

ответ \_\_\_\_\_

78. Какое ключевое слово используется для указания того, что класс реализует интерфейс?

ответ \_\_\_\_\_

79. Какое название получил механизм, позволяющий JVM работать с примитивными типами данных как с объектами?

ответ \_\_\_\_\_

80. Какой вид будет иметь инструкция, добавляющая в область видимости описываемого класса все классы из пакета test?

ответ \_\_\_\_\_

81. Каким образом будет выглядеть инструкция, указывающая на принадлежность описываемого класса к пакету test, который в свою очередь является подпакетом класса biggest?

ответ \_\_\_\_\_

82. Какой вид будет иметь инструкция, добавляющая в область видимости описываемого класса класс test из пакета test?

ответ \_\_\_\_\_

83. Какой вид будет иметь инструкция, указывающая на принадлежность описываемого класса к пакету test?

ответ \_\_\_\_\_

84. С помощью какой команды может быть создан вектор с именем test из элементов типа TestE1?

ответ \_\_\_\_\_

85. С помощью какой команды может быть создан массив с именем test из десяти элементов типа int?

ответ \_\_\_\_\_

86. Как будет выглядеть команда для запуска класса Test, готовый байт код которого содержится в файле Test.class

ответ \_\_\_\_\_

87. Как называется организация, регулирующая развитие UML? ·

OMG ·

ANSI ·

ACM ·

NASA

88. Что из перечисленного может быть отнесено к членам класса? ·

Вложенный класс ·

Метод ·

Объект ·

Поле

89. Что из перечисленного может быть отнесено к преимуществам шаблона «шаблонный метод»? ·

Помогает избегать дублирования повторяющихся конструкций ·

Позволяет динамически варьировать поведение системы ·

Позволяет оптимизировать передачу данных при вызове методов ·



Не требует дополнительной иерархии классов

Методические указания по проведению тестирования.

Студенту в случайном порядке предлагается ответить на 30 вопросов. Время для ответа – 45 минут.

Критерии оценивания:

100 баллов – при правильном ответе на 30 вопросов.

85-99 баллов – при правильном ответе на 25-29 вопросов.

70-84 балла – при правильном ответе на 20-24 вопроса.

65-69 баллов – при правильном ответе на 15-19 вопросов.

0-64 баллов – при правильном ответе на ответе на 14 и менее вопросов

Количество баллов	0...64	65...74	75...84	85...100
Шкала оценивания	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично
	Не зачтено		Зачтено	

#### 2.4 Оценочные средства при промежуточной аттестации

Для промежуточной аттестации используется стандартный экзаменационный подход с использованием теоретических и практических заданий (перечень теоретических экзаменационных вопросов предоставлен заранее, практические задания предоставляются непосредственно к моменту их исполнения).

Формами промежуточной аттестации являются зачет и экзамен, в процессе которых определяется сформированность обозначенных в рабочей программе компетенций. Инструментом измерения сформированности компетенций являются оформленные и зачтенные отчеты по лабораторным работам, ответы на вопросы во время опроса.

Зачет проводится в форме контрольного мероприятия с опросом по зачтенным отчетам по лабораторным работам, по темам лекций.

Обучающийся отвечает на два вопроса.

Оценка выставляется с учетом отчетов по лабораторным работам и ответа на вопросы.

Критерии оценивания:

- 100 баллов – при правильном и полном ответе на два вопроса;
- 75...99 баллов – при правильном и полном ответе на один из вопросов и правильном, но не полном ответе на другой из вопросов;
- 50...74 баллов – при правильном и неполном ответе на два вопроса или правильном и полном ответе только на один из вопросов;
- 25...49 баллов – при правильном и неполном ответе только на один из вопросов; - 0...24 баллов – при отсутствии правильных ответов на вопросы.

Количество баллов	0-49	50-100
Шкала оценивания	Не зачтено	Зачтено

Примеры вопросов:

1. Продемонстрируйте преимущество подхода объектной декомпозиции в программировании.
2. Опишите методики повторного использования программного кода, примененные в ходе выполнения лабораторных работ.

На экзамене обучающийся отвечает на билет, в котором содержится 2 вопроса. Оценка за экзамен выставляется с учетом отчетов по лабораторным работам и ответа на вопросы.

Критерии оценивания:

- 100 баллов – при правильном и полном ответе на два вопроса;
- 75...99 баллов – при правильном и полном ответе на один из вопросов и правильном, но не полном ответе на другой из вопросов;
- 50...74 баллов – при правильном и неполном ответе на два вопроса или правильном и полном ответе только на один из вопросов;
- 25...49 баллов – при правильном и неполном ответе только на один из вопросов;
- 0...24 баллов – при отсутствии правильных ответов на вопросы.

Шкала оценивания:

Количество баллов	0-64	65-74	75-84	85-100
Шкала оценивания	Неуд.	Удовл.	Хор.	Отл.

**Тематика экзаменационных вопросов:**

1. Основы ООП (общие термины)
2. Конструирование объектов
3. Принципы декомпозиции при объектно-ориентированном построении
4. Объектная декомпозиция: объекты и сообщения
5. Работа с файлами: потоковый ввод/вывод
6. Работа с файлами: методы
7. Атрибуты
8. Типы параметров методов
9. Шаблонные классы
10. События и делегаты. Реакция на события
11. Обработка исключений. Блоки исключений, типы исключений
12. Работа с XML-файлами: DOM
13. Работа с XML-файлами: SAX
14. Использование агрегирования: агрегация и композиция
15. ООПиРИС: цели и задачи изучения дисциплины
16. Жизненный цикл программных систем: понятие, основные процессы
17. Хранение коллекций однотипных объектов
18. Интерфейсные типы
19. Назначение ключевого слова base
20. Назначение ключевого слова this
21. Класс. Объект
22. Методы
23. Поля
24. Свойства
25. Неуправляемые ресурсы
26. Шаблон проектирования MVC
27. Проектирование пользовательского интерфейса
28. Инкапсуляция
29. Наследование
30. Полиморфизм
31. Директивы области видимости: public, internal
32. Директивы области видимости: private, protected
33. Rapid Application Development (RAD)
34. Управление изменениями кода: системы контроля версий
35. Статические методы
36. Виртуальные методы
37. Абстрактные методы

38. Статические методы
39. Статические классы
40. Общая характеристика и возможности среды разработки Microsoft Visual Studio

## **2.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций**

При проведении текущего контроля успеваемости в форме опроса по распоряжению педагогического работника обучающиеся убирают все личные вещи, электронные средства связи, печатные и (или) рукописные источники информации, достают чистый лист бумаги любого размера и ручку. На листе бумаги записываются Фамилия, Имя, Отчество (при наличии), номер учебной группы и дата проведения текущего контроля успеваемости. Педагогический работник задает вопросы, которые могут быть записаны на подготовленный для ответа лист бумаги. В течение установленного педагогическим работником времени обучающиеся письменно формулируют ответы на заданные вопросы. По истечении установленного времени лист бумаги с подготовленными ответами обучающиеся передают педагогическому работнику для последующего оценивания результатов текущего контроля успеваемости.

Результаты текущего контроля успеваемости доводятся до сведения обучающихся в течение трех учебных дней, следующих за днем проведения текущего контроля успеваемости, и могут быть учтены педагогическим работником при промежуточной аттестации. Результаты промежуточной аттестации доводятся до сведения обучающихся в день проведения промежуточной аттестации. При подготовке ответов на вопросы при проведении текущего контроля успеваемости и при прохождении промежуточной аттестации обучающимся запрещается использование любых электронных средств связи, печатных и (или) рукописных источников информации. В случае обнаружения педагогическим работником факта использования обучающимся при подготовке ответов на вопросы указанных источников информации – оценка результатов текущего контроля успеваемости и (или) промежуточной аттестации соответствует 0 баллов.

При прохождении текущего контроля успеваемости и промежуточной аттестации обучающимися с ограниченными возможностями здоровья и инвалидами, допускается присутствие в помещении лиц, оказывающим таким обучающимся соответствующую помощь, а для подготовки ими ответов отводится дополнительное время с учетом особенностей их психофизического развития, индивидуальных возможностей и состояния здоровья.