

Приложение

 к рабочей программе дисциплины

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**Объектно-ориентированное программирование**

 *(наименование дисциплины (модуля)*

Направление подготовки / специальность

**09.03.01 Информатика и вычислительная техника**

*(код и наименование)*

Направленность (профиль) / специализация

«Проектирование АСОИУ на транспорте»

*(наименование)*

Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

**1. Пояснительная записка**

 Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации:

* ***зачет 2 семестр***
* ***курсовая работа 3 семестр***
* ***экзамен 3 семестр.***

Перечень компетенций, формируемых в процессе освоения дисциплины

|  |  |
| --- | --- |
| Код и наименование компетенции | Код индикатора достижения компетенции |
| ПК-1: Способен разрабатывать требования и проектировать программное обеспечение | ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня |

Результаты обучения по дисциплине, соотнесенные с планируемыми

результатами освоения образовательной программы

|  |  |  |
| --- | --- | --- |
| Код и наименование индикатора достижения компетенции | Результаты обучения по дисциплине | Оценочные материалы (семестр\_\_) |
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня  | Обучающийся знает: объектно-ориентированное программирование | Вопросы (№1–№5) |
| Обучающийся умеет: разрабатывать и реализовывать классы объектов | Задания (№1–№3) |
| Обучающийся владеет: современными средствами разработки объектно-ориентированных систем | Задания (№1–№3) |
| ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня | Обучающийся знает: общие принципы конструирования программ с использованием объектно-ориентированной парадигмы | Вопросы (№6–№10) |
| Обучающийся умеет: использовать визуальную среду программирования | Задания (№4–№6) |
| Обучающийся владеет: объектной декомпозицией и проектированием; | Задания (№4–№6) |

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

1. Собеседование
2. выполнение заданий в ЭИОС СамГУПС.

Промежуточная аттестация (курсовая работа работа) проводится в одной из следующих форм:

1. Собеседование
2. выполнение заданий в ЭИОС СамГУПС.

Промежуточная аттестация (Экзамен) проводится в одной из следующих форм

1) Собеседование

2) выполнение заданий в ЭИОС СамГУПС.

**2. Типовые[[1]](#footnote-2) контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций**

**2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата**

Проверяемый образовательный результат:

|  |  |
| --- | --- |
| Код и наименование индикатора достижения компетенции | Образовательный результат |
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня | Обучающийся знает: объектно-ориентированное программирование |
| *Вопросы:* 1. Какие требования должны учитываться при разработке компонентных объектов?
* четкая специализация компонента с точки зрения конкретной решаемой задачи
* возможность взаимодействия с другими компонентами
* возможность использования компонентов в инструментах быстрой разработки приложений
* соблюдение правил используемой при разработке компонентной модели
1. Какие утверждения справедливы относительно механизма сериализации?
* сериализация позволяет сохранять объекты в файлах в виде потока байтов
* сериализация применима для объектов любых классов, которые реализуют интерфейс Serializable
* сериализация позволяет сохранять и восстанавливать объектную структуру любой сложности
* сериализация основана на использовании механизма позднего связывания
1. Какие преимущества дает использование компонентного подхода при разработке приложений?
* сокращение объема кода, написанного «вручную»
* уменьшение сроков разработки приложений
* получение более быстрого программного кода
* возможность максимально полного использования аппаратных особенностей процессоров
1. Какие шаги включает в себя процесс разработки оконных приложений с помощью стандартных компонентов?
* размещение необходимых компонентов на форме
* установка свойств размещенных на форме компонентов
* написание необходимых обработчиков событий
* разработка используемых компонентных классов
1. Какие утверждения справедливы относительно понятия «компонент»?
* компонент — это объект специального вида
* компоненты являются экземплярами компонентных классов
* компоненты являются основой средств быстрой разработки приложений
* компоненты используются только для создания оконного пользовательского интерфейса
 |
| ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня | Обучающийся знает: общие принципы конструирования программ с использованием объектно-ориентированной парадигмы |
| *Вопросы:*1. Какие утверждения относительно методов-конструкторов являются правильными?
* конструктор выполняет инициализацию свойств объекта
* конструктор вызывается раньше всех остальных методов
* конструктор отвечает за создание объекта при выполнении программы
* конструктор отвечает за освобождение памяти, выделенной объекту
1. Какие типы свойств может содержать объект?
* основные стандартные базовые типы
* структурированные свойства-массивы
* объектные свойства
* программные свойства
1. Какие типы методов обычно содержат классы?
* конструкторы
* методы доступа к свойствам
* методы, реализующие функциональность объектов класса
* деструкторы
1. Какие утверждения относительно понятия «Программный объект» являются правильными?
* с каждым объектом связываются данные и программный код
* программный объект является моделью исходного объекта
* объект существует при выполнении программы, занимая часть оперативной памяти
* объекты используются для описания только физических сущностей
1. Какие утверждения относительно методов-деструкторов являются правильными?
* деструктор отвечает за освобождение памяти, выделенной объекту
* деструкторы реализованы не во всех объектных языках
* в языках Java и С# вместо деструкторов используется механизм сборки мусора
* деструктор отсутствуют в языке С++
 |

**2.2 Типовые задания для оценки навыкового образовательного результата**

Проверяемый образовательный результат:

|  |  |
| --- | --- |
| Код и наименование индикатора достижения компетенции | Образовательный результат |
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня | Обучающийся умеет: разрабатывать и реализовывать классы объектовОбучающийся владеет: современными средствами разработки объектно-ориентированных систем |
| *Пример задания:*1. *Декомпозировать прикладную задачу под объектно-ориентированный подход*
2. *Разработать программу средствами языка Си++ в объектной парадигме*
3. *Разработать программу* используя классы и алгоритмы библиотеки STL.
 |
| ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня | Обучающийся умеет: использовать визуальную среду программированияОбучающийся владеет: объектной декомпозицией и проектированием; |
| *Пример задания*1. *Программирование вычислительных процессов с использованием рекурсивных алгоритмов*
2. *Разработать классы с учетом обработки исключительных ситуаций*
3. *Разработать параметризованные классы (шаблоны классов). Перегруженные функции и функции-шаблоны.*
 |

**2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации**

**Вопросы к зачету:**

1. Общая характеристика и основные принципы Объектно-ориентированного программирования.

2. Определение класса в C++. Понятие объектов.

3. Функциональные компоненты класса. Типы функциональных компонент.

4. Конструкторы.

5. Деструкторы.

6. Перегрузка функциональных элементов и операций.

7. Константные объекты и функции. Ключевое слово this.

8. Наследование классов. Конструкторы и деструкторы при наследовании.

9. Виртуальные функции. Понятие полиморфизма и позднего связывания.

10. Неоднозначность при множественном наследовании. Виртуальный базовый класс.

11. Абстрактный класс и чистые виртуальные функции.

12. Дружественные функции и классы.

13. Ссылки.

14. Потоковые классы. Операции извлечения и помещения данных.

15. Форматирование данных. Флаги и форматирующие методы.

16. Манипуляторы.

17. Методы обмена с потоками.

18. Файловые потоки.

19. Ошибки потоков.

20. Обработка исключительных ситуаций. Понятие исключений. Общий механизм обработки исключений.

21. Синтаксис исключений. Перехват исключений.

22. Списки исключений функций. Исключения в конструкторах и деструкторах.

23. Стандартные исключения. Иерархии исключений.

24. Шаблоны функций.

**Вопросы к экзамену:**

1. Общая характеристика и основные принципы Объектно-ориентированного программирования.

2. Определение класса в C++. Понятие объектов.

3. Функциональные компоненты класса. Типы функциональных компонент.

4. Конструкторы.

5. Деструкторы.

6. Перегрузка функциональных элементов и операций.

7. Константные объекты и функции. Ключевое слово this.

8. Наследование классов. Конструкторы и деструкторы при наследовании.

9. Виртуальные функции. Понятие полиморфизма и позднего связывания.

10. Неоднозначность при множественном наследовании. Виртуальный базовый класс.

11. Абстрактный класс и чистые виртуальные функции.

12. Дружественные функции и классы.

13. Классы потоков языка C++. Операции извлечения и помещения данных.

14. Способы форматирования при работе с классами потоков. Форматирующие функции.

15. Флаги форматирования.

16. Манипуляторы.

17. Ошибки потоков.

18. Файловый ввод-вывод с применением потоков. Открытие файла.

19. Ввод-вывод в файлы с использованием потоков.

20. Строко - ориентированный ввод-вывод.

21. Неформатированный ввод-вывод.

22. Пространство имен.

23. Обработка исключительных ситуаций. Понятие исключений. Общий механизм обработки исключений.

24. Синтаксис исключений. Перехват исключений.

25. Списки исключений функций. Исключения в конструкторах и деструкторах.

26. Стандартные исключения. Иерархии исключений.

27. Шаблоны функций. Использование шаблонов функций.

28. Специализированная функция шаблона. Перегрузка шаблонов.

29. Определение и использование шаблонов классов.

30. Использование в шаблонных классах аргументов по умолчанию. Специализация шаблонов классов. Достоинства и недостатки шаблонов.

31. Класс string. Конструкторы. Преобразование строк.

32. Класс string. Поиск подстрок. Сравнение частей строк.

33. Класс string. Получение количества элементов. Изменение размера стро

34. Библиотека шаблонов STL.Назначение STL и состав библиотеки.

35. Понятие контейнера и их типы. итератора. Поля контейнеров.

36. Итераторы. Их типы. Методы для просмотра контейнеров с помощью итераторов.

37. Последовательные контейнеры. Общие операции для работы с контейнерами.

38. Контейнерный класс vector. Его конструкторы. Операции присваивания и копирования. Доступ к элементам вектора.

39. Контейнерный класс vector. Его методы capacity, reserve, resize. Методы для изменения объектов класса vector.

40. Векторы логических значений.

41. Двухсторонние очереди (deque). Конструкторы. Операции, которые реализованы в deque. Методы добавления и выборки.

42. Списки list. Понятие и реализация. Доступ к элементам. Занесение в начало и конец.

43. Методы изменения объектов списка. Сцепление списков. Удаление элементов. Сортировка.

44. Адаптеры стеки и очереди.

**3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации**

**Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий**

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90% от общего объёма заданных вопросов;

- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76% от общего объёма заданных вопросов;

- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы –75–60 % от общего объёма заданных вопросов;

- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60% от общего объёма заданных вопросов.

**Критерии формирования оценок по результатам выполнения заданий**

**«Отлично/зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов.

**«Хорошо/зачтено»** – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

**«Удовлетворительно/зачтено»** – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

**«Неудовлетворительно/не зачтено»** – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

*Виды ошибок:*

*- грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

*- негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*

*- недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

**Критерии формирования оценок по зачету**

**«Зачтено»** - обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности.

**«Не зачтено»** - выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.

**Критерии формирования оценок по написанию и защите курсовой работы**

**«Отлично»** (5 баллов) – получают обучающиеся, оформившие курсовую работу (курсовой проект) в соответствии с предъявляемыми требованиями, в которой отражены все необходимые результаты проведенного анализа, сделаны обобщающие выводы и предложены рекомендации в соответствии с тематикой курсовой работы, а также грамотно и исчерпывающе ответившие на все встречные вопросы преподавателя.

**«Хорошо» (**4 балла) – получают обучающиеся, оформившие курсовую работу (курсовой проект) в соответствии с предъявляемыми требованиями, в которой отражены все необходимые результаты проведенного анализа, сделаны обобщающие выводы и предложены рекомендации в соответствии с тематикой курсовой работы. При этом при ответах на вопросы преподавателя обучающийся допустил не более двух ошибок.

**«Удовлетворительно»** (3 балла) – получают обучающиеся, оформившие курсовую работу (курсовой проект) в соответствии с предъявляемыми требованиями. При этом при ответах на вопросы преподавателя обучающийся допустил более трёх ошибок.

**«Неудовлетворительно» (0 баллов) –** ставится за курсовую работу, если число ошибок и недочетов превысило удовлетворительный уровень компетенции

**Критерии формирования оценок по экзамену**

**«Отлично»** (5 баллов) – обучающийся демонстрирует знание всех разделов изучаемой дисциплины: содержание базовых понятий и фундаментальных проблем; умение излагать программный материал с демонстрацией конкретных примеров. Свободное владение материалом должно характеризоваться логической ясностью и четким видением путей применения полученных знаний в практической деятельности, умением связать материал с другими отраслями знания.

**«Хорошо»** (4 балла) – обучающийся демонстрирует знания всех разделов изучаемой дисциплины: содержание базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности. Таким образом данная оценка выставляется за правильный, но недостаточно полный ответ.

**«Удовлетворительно»** (3 балла) – обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. Однако знание основных проблем курса не подкрепляются конкретными практическими примерами, не полностью раскрыта сущность вопросов, ответ недостаточно логичен и не всегда последователен, допущены ошибки и неточности.

**«Неудовлетворительно»** (0 баллов) – выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.

1. Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств. [↑](#footnote-ref-2)