

Приложение 2 к РПД
Объектно-ориентированное программирование
01.03.02 Информатика и вычислительная техника
Направленность (профиль)
Виртуальные технологии и дизайн
Форма обучения – очная
Год набора – 2020

**ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

1.	Кафедра	Математики, физики и информационных технологий
2.	Направление подготовки	09.03.01 Информатика и вычислительная техника
3.	Направленность (профиль)	Виртуальные технологии и дизайн
4.	Дисциплина (модуль)	Б1.О.14.04 Объектно-ориентированное программирование
5.	Форма обучения	очная
6.	Год набора	2021

1. Перечень компетенций

- | |
|--|
| – ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения |
|--|

Критерии и показатели оценивания компетенций на различных этапах их формирования

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
Объектно-ориентированный подход как парадигма программирования и проектирования программного обеспечения	ОПК-8	этапы решения задач на ЭВМ и жизненного цикла программного продукта; место парадигмы ООП среди других парадигм программирования; роль ООП при создании приложений с графическим интерфейсом; область применения парадигмы ООП; основные принципы работы в средах разработки ООП;	применять технологии структурного и процедурного программирования; выполнять трассировку программного кода;	понятийным аппаратом ООП и методикой моделирования абстракций с описанием свойств, поведения и отношений между абстракциями	Тестирование, Лабораторные работы, Доклад/реферат, Презентация
Основные принципы ООП и их реализация	ОПК-8	модель выделяемой приложению памяти и управление ею в случае языка, поддерживающего ООП (статический раздел, стек, куча), назначение ссылок и указателей; понятия абстрактный тип данных и структура данных; теоретико-множественное определение понятия класс, соотношение понятий тип данных, пользовательский тип данных, перечисление, структура, класс; связь понятий сущность, класс, экземпляр; типизацию конструкторов класса; основные принципы объектно-ориентированного программирования; способы реализации этих принципов средствами хотя бы одного языка программирования (C++ в данном курсе); особенности поддержки ООП языком C++ в сравнении с C#, Java, Python (для последних версий стандартов языка).	использовать инструментарий среды разработки для организации работы с классами; создавать собственный класс; формализовать описание сущности (её свойств и поведения) в виде программного кода парадигмы ООП; создавать иерархии классов; реализовывать различные виды полиморфизма;	технологией создания простейших приложений средствами инструментальных сред разработки, сборки, отладки, запуска; навыками программирования в современных средах, поддерживающих ООП; навыками формализации задачи на естественном языке с использованием понятийного аппарата ООП: сущность, экземпляр сущности, отношение между сущностями (классами/объектами), взаимодействие между сущностями, передача управления между сущностями навыками написания программного кода с учётом разделения определения, реализации и использования классов; CASE-средствами создания диаграмм классов;	Тестирование, Лабораторные работы, Доклад/реферат, Презентация
Контейнеры. Шаблоны. Обработка исключений. Паттерны проектирования	ОПК-8	определения и типизацию для понятий: контейнеры, шаблоны, алгоритмы и функциональные объекты библиотеки шаблонов, паттерны проектирования	работать с шаблонами STL; использовать контейнеры; использовать алгоритмы и функциональные объекты библиотеки STL; порождать и обрабатывать исключения в программном коде; применять паттерны проектирования	навыками программирования в современных ООП средах; использования методов ООП при решении задач различного уровня сложности; технологией разработки, тестирования, документирования и сопровождения программного продукта.	Тестирование, Лабораторные работы, Доклад/реферат, Презентация

Этап формирования	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности
		Знать:	Уметь:	Владеть:	
Разработка графического пользовательского интерфейса средствами ООП	ОПК-8	понятие графического пользовательского интерфейса (ГПИ); состав и назначение отдельных элементов ГПИ; состав и назначение стандартных библиотек для разработки ГПИ;	создавать стандартные оконные приложения с графическим пользовательским интерфейсом;	навыками создания приложений с графическим интерфейсом (desktopных оконных приложений)	Тестирование, Лабораторные работы, Доклад/реферат, Презентация

Шкала оценивания в рамках балльно-рейтинговой системы

«неудовлетворительно» – 60 баллов и менее; «удовлетворительно» – 61-80 баллов; «хорошо» – 81-90 баллов; «отлично» – 91-100 баллов

4. Критерии и шкалы оценивания

4.1. Тест по разделу дисциплины

Процент правильных ответов	До 60	61-80	81-100
Количество баллов за решенный тест	0	6-8	9-10

4.2. Критерии оценки выполнения лабораторной работы

Баллы	Характеристики ответа студента
1	Лабораторная работа выполнена не менее чем на 91%
0,8	Лабораторная работа выполнена не менее чем на 81%
0,6	Лабораторная работа выполнена не менее чем на 61%
0	Лабораторная работа выполнена менее чем на 60%

4.3. Презентация

Структура презентации	Максимально е количество баллов
Содержание	
Сформулирована цель работы	1
Понятны задачи и ход работы	1
Информация изложена полно и четко	1
Иллюстрации усиливают эффект восприятия текстовой части информации	1
Сделаны выводы	1
Оформление презентации	
Единый стиль оформления	1
Текст легко читается, фон сочетается с текстом и графикой	1
Все параметры шрифта хорошо подобраны, размер шрифта оптимальный и одинаковый на всех слайдах	1
Ключевые слова в тексте выделены	1
Эффект презентации	
Общее впечатление от просмотра презентации	1
Мах количество баллов	10
Окончательная оценка:	

4.4. Доклад/Реферат

Баллы	Характеристики ответа студента
10	<ul style="list-style-type: none">– студент глубоко и всесторонне усвоил проблему;– уверенно, логично, последовательно и грамотно его излагает;– опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью;– умело обосновывает и аргументирует выдвигаемые им идеи;– делает выводы и обобщения;– свободно владеет понятиями.
8	<ul style="list-style-type: none">– студент твердо усвоил тему, грамотно и по существу излагает ее, опираясь на знания основной литературы;– не допускает существенных неточностей;– увязывает усвоенные знания с практической деятельностью;– аргументирует научные положения;– делает выводы и обобщения;– владеет системой основных понятий.
6	<ul style="list-style-type: none">– тема раскрыта недостаточно четко и полно, то есть студент освоил проблему, по существу излагает ее, опираясь на знания только основной литературы;– допускает несущественные ошибки и неточности;– испытывает затруднения в практическом применении знаний;

Баллы	Характеристики ответа студента
	<ul style="list-style-type: none"> – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой понятий.
0	<ul style="list-style-type: none"> – студент не усвоил значительной части проблемы; – допускает существенные ошибки и неточности при рассмотрении ее; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений; – не владеет понятийным аппаратом.

5. Типовые контрольные задания и методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

5.1 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

5.2 Вопросы к экзамену:

1. Базовые понятия объектно-ориентированного программирования.
2. Объектно-ориентированное программирование: 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.