

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:
Первый проректор
_____ / Т.Р. Змызгова/
« ____ » _____ 2024 г.

Рабочая программа учебной дисциплины
**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИ-
РОВАНИЕ**

образовательной программы высшего образования –
программы бакалавриата

09.03.04 – Программная инженерия

Направленность:
Программная обеспечение автоматизированных систем

Форма обучения: очная и заочная

Курган 2024

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с учебными планами по программе бакалавриата «Программная инженерия» (Программное обеспечение автоматизированных систем), утвержденными для очной формы обучения «28» июня 2024 года, для заочной формы обучения «28» июня 2024 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» «29» августа 2024 года, протокол № 1.

Рабочую программу составили:

Доцент кафедры
«Программное обеспечение
автоматизированных систем»,
к.ф.-м.н, доцент

О.С. Черепанов

Согласовано:

Заведующий кафедрой
«Программное обеспечение
автоматизированных систем»
к.ф.-м.н., доцент

С. В. Косовских

Начальник управления
образовательной деятельности

И. В. Григоренко

Специалист по учебно-методической
работе учебно-методического отдела

Г.В. Казанкова

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 4 зачетных единиц трудоемкости (144 академических часов)

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр	
		3	4
Аудиторные занятия (контактная работа с преподавателем), всего часов	48	40	8
в том числе:			
Лекции	16	16	-
Лабораторные работы	24	24	-
Практические работы	8	-	8
Аудиторные занятия в интерактивной форме, часов	-	-	-
Самостоятельная работа, всего часов	96	32	64
в том числе:			
Подготовка к зачету	18	18	-
Подготовка к экзамену	27	-	27
Курсовая работа	36	-	36
Другие виды самостоятельной работы	15	14	1
Вид промежуточной аттестации	зачет/экзамен	экзамен	зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	144	72	72

Заочная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		4
Аудиторные занятия (контактная работа с преподавателем), всего часов	12	12
в том числе:		
Лекции	4	4
Лабораторные работы	8	8
Аудиторные занятия в интерактивной форме, часов	-	-
Самостоятельная работа, всего часов	132	132
в том числе:		
Подготовка к зачету	18	18
Курсовая работа	36	36
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	78	78
Вид промежуточной аттестации	зачет	зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	144	144

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Объектно-ориентированное программирование» относится к Блоку 1 обязательной части образовательной программы дисциплин модуля «Информатика и программирование».

Изучение дисциплины базируется на результатах обучения, сформированных при изучении следующих дисциплин:

- Информатика.
- Основы программирования.
- Алгоритмы и структуры данных.

Результаты обучения по дисциплине необходимы для изучения дисциплин: «Технологии разработки Web-приложений», «Архитектура информационных систем», «Разработка мобильных приложений», «Задачи и методы искусственного интеллекта», «Прикладные задачи машинного обучения», «Нейрокомпьютерные системы» и выполнения выпускной квалификационной работы.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Целью освоения дисциплины является формирование теоретических знаний и практических навыков применения объектно-ориентированной парадигмы для реализации программных систем.

Задачами дисциплины являются изучение теоретических понятий, лежащих в основе процесса объектно-ориентированной разработки программного обеспечения, а также освоение современных технологий ООП.

Компетенции, формируемые в результате освоения дисциплины:

- способность использовать современные информационные технологии и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности (ОПК-2);

- способность разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-6);

- способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-8).

Индикаторы и дескрипторы части соответствующей компетенции, формируемой в процессе изучения дисциплины «Объектно-ориентированное программирование», оцениваются при помощи оценочных средств.

Планируемые результаты обучения по дисциплине «Объектно-ориентированное программирование», индикаторы достижения компетенций
ОПК-2, ОПК-6, ОПК-8 перечень оценочных средств

№ п/п	Код индикатора достижения компетенции	Наименование индикатора достижения компетенции	Код планируемого результата обучения	Планируемые результаты обучения	Наименование оценочных средств
1	ИД-1ОПК-2	<i>Знать:</i> основные концепции объектно-ориентированного программирования	З (ИД-1ОПК-2)	<i>Знает:</i> основные концепции объектно-ориентированного программирования	Вопросы для сдачи зачета и экзамена
2	ИД-2ОПК-2	<i>Уметь:</i> применять основные концепции объектно-ориентированного программирования	У (ИД-1ОПК-2)	<i>Умеет:</i> применять основные концепции объектно-ориентированного программирования	Вопросы для сдачи зачета и экзамена
3	ИД-3ОПК-2	<i>Владеть:</i> опытом применения основных концепций объектно-ориентированного программирования	В (ИД-2ОПК-2)	<i>Владеет:</i> опытом применения основных концепций объектно-ориентированного программирования	Вопросы для сдачи зачета и экзамена
4	ИД-1ОПК-6	<i>Знать:</i> программные средства реализации графического пользовательского интерфейса	З (ИД-1ОПК-6)	<i>Знает:</i> программные средства реализации графического пользовательского интерфейса	Вопросы для сдачи зачета и экзамена
5	ИД-2ОПК-6	<i>Уметь:</i> использовать программные средства реализации графического пользовательского интерфейса	У (ИД-2ОПК-6)	<i>Умеет:</i> использовать программные средства реализации графического пользовательского интерфейса	Вопросы для сдачи зачета и экзамена
6	ИД-3ОПК-6	<i>Владеть:</i> опытом использования программных средств реализации графического пользовательского интерфейса	В (ИД-3ОПК-6)	<i>Владеет:</i> опытом использования программных средств реализации графического пользовательского интерфейса	Вопросы для сдачи зачета и экзамена
7	ИД-1ОПК-8	<i>Знать:</i> объектно-ориентированные языки программирования	З (ИД-1ОПК-8)	<i>Знает:</i> объектно-ориентированные языки программирования	Вопросы для сдачи зачета и экзамена
8	ИД-2ОПК-8	<i>Уметь:</i> реализовать программное обеспечение с использованием парадигмы ООП	У (ИД-2ОПК-8)	<i>Умеет:</i> реализовать программное обеспечение с использованием парадигмы ООП	Вопросы для сдачи зачета и экзамена
9	ИД-3ОПК-8	<i>Владеть:</i> современными объектно-ориентированными языками программирования	В (ИД-3ОПК-8)	<i>Владеет:</i> современными объектно-ориентированными языками программирования	Вопросы для сдачи зачета и экзамена

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план

Очная форма обучения (3 семестр)

№	Наименование раздела	Количество часов контактной работы с преподавателем	
		Лекции	Лабораторные работы
1	Теоретические концепции объектно-ориентированных языков программирования	4	-
2	Реализация ООП в языка Python	6	10
	Рубежный контроль №1		2
3	Реализация ООП в языке C++	6	10
	Рубежный контроль №2		2
Всего:		16	24

Очная форма обучения (4 семестр)

№	Наименование раздела	Количество часов контактной работы с преподавателем
		Практические работы
4	Разработка приложений с графическим пользовательским интерфейсом	8
Всего:		8

Заочная форма обучения

№	Наименование раздела	Количество часов контактной работы с преподавателем	
		Лекции	Лабораторные работы
1	Теоретические концепции объектно-ориентированных языков программирования	-	-
2	Реализация ООП в языка Python	2	4
3	Реализация ООП в языке C++	2	4
Всего:		4	8

4.2. Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем	
	Очная форма	Заочная форма

Раздел 1. Теоретические концепции объектно-ориентированных языков программирования

<p>Лекция 1. История возникновения и развития объектно-ориентированной парадигмы программирования</p> <p>История возникновения ООП. Проблемы программного обеспечения. Виды декомпозиции задач. История развития объектно-ориентированных языков программирования.</p>	1	
<p>Лекция 2. Объектная модель. Классы и объекты</p> <p>Основные положения объектной модели. Составные части объектного подхода: абстракция, инкапсуляция, модульность, иерархия, контроль типов.</p> <p>Понятие объекта. Состояние объекта. Поведение объекта. Операции. Роли и обязанности объекта. Отношения между объектами. Понятие класса. Интерфейс и реализация. Отношение между классами. Полиморфизм. Взаимосвязь объектов и классов. Качество классов и объектов.</p>	3	0
<h3>Раздел 2. Реализация ООП в языке Python</h3>		
<p>Лекция 3. Классы и объекты</p> <p>Определение класса. Атрибуты класса. Создание экземпляров класса. Алгоритм поиска атрибутов. Атрибуты класса. Методы класса и статические методы. Абстрактные методы.</p>	2	
<p>Лекция 4. Наследование</p> <p>Магические методы (операторы). Наследование классов. Переопределение методов. Множественное наследование. Миксины.</p>	2	2
<p>Лекция 5. Исключения</p> <p>Обработка исключений в Python. Стандартные типы исключений. Контролируемый блок. Оператор raise. Обработчики исключений. Параметры обработчиков исключений. Алгоритм поиска обработчиков исключений.</p>	2	
<h3>Раздел 3. Реализация ООП в языке C++</h3>		
<p>Лекция 6. Классы и объекты</p>	2	

<p>История создания языка C++. Классы в языке C++. Спецификаторы доступа. Конструкторы. Деструкторы. Определение метода вне класса. Перегрузка операторов. Преобразование типов. Константные методы. Статические свойства и методы. Вложенные классы. Дружественность.</p>		
<p>Лекция 7. Наследование</p> <p>Наследование. Базовые и производные классы. Доступ к базовому классу. Конструкторы при наследовании. Перегрузка методов. Иерархия классов. Доступ к объектам иерархии. Виртуальные методы. Абстрактные классы. Множественное наследование.</p>	2	2
<p>Лекция 8. Исключения и шаблоны классов</p> <p>Обработка исключений в C++. Контролируемый блок. Оператор throw. Обработчики исключений. Параметры обработчиков исключений. Алгоритм поиска обработчиков исключений.</p> <p>Шаблоны. Шаблоны методов. Шаблоны классов. Инстанцирование. Шаблоны классов при наследовании.</p>	2	
<p>Итого:</p>	16	4

4.3. Лабораторные работы

Номер раздела, темы	Наименование раздела	Наименование лабораторной работы	Норматив времени, час.	
			Очная форма обучения	Заочная форма обучения
2	Реализация ООП в языке Python	Лабораторная работа №1. Простой класс.	10	4
		Лабораторная работа №2. Полноценные классы со свойствами и методами.		
Лабораторная работа №3. Работа с операторами.				
Лабораторная работа №4. Отношение между классами.				
Лабораторная работа №5. Построение иерархии классов. Одиночное наследование.				
Лабораторная работа №6. Множественное наследование.				
Лабораторная работа №7. Интерфейсы.				
Лабораторная работа №8. Исключения.				
		Рубежный контроль №1	2	-

3	Реализация ООП в языке C++	Лабораторная работа №9. Простой класс.	10	4
		Лабораторная работа №10. Полноценные классы со свойствами и методами.		
		Лабораторная работа №11. Работа с операторами.		
		Лабораторная работа №12. Отношение между классами.		
		Лабораторная работа №13. Построение иерархии классов. Одиночное наследование.		
		Лабораторная работа №14. Множественное наследование.		
		Лабораторная работа №15. Интерфейсы.		
		Лабораторная работа №16. Исключения.		
		Рубежный контроль №2	2	-
Всего:			24	8

4.4. Практические работы (очная форма)

Номер раздела, темы	Наименование раздела	Наименование практической работы	Норматив времени, час.
4	Разработка приложений с графическим пользовательским интерфейсом	<i>Практическая работа №1</i> Введение в Qt. Обзор иерархии классов. Элементы управления размещением элементов	2
		<i>Практическая работа №2</i> Элементы отображения (Надписи, индикаторы). Кнопки, флажки и переключатели	
		<i>Рубежный контроль №3</i>	2

		<i>Практическая работа №3</i> Элементы ввода и выбора, отображения. События	2
		<i>Практическая работа №4</i> Графика и звук	
		<i>Рубежный контроль №4</i>	2
Всего:			8

4.5 Курсовая работа

В процессе выполнения курсовой работы обучающийся самостоятельно разрабатывает программное обеспечение в соответствии с индивидуальным заданием, согласованным и утвержденным в установленном порядке.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Во время лекций по дисциплине обучающимся рекомендуется конспектировать теоретический материал, отмечая важные моменты, на которые заострил внимание преподаватель, участвовать в опросах и дискуссиях. Перед лекцией необходимо повторить выданный материал, зафиксировать непонятные моменты, чтобы обсудить их на занятии. Конспект лекций представлен в виде мультимедийных презентаций и включен в состав методического комплекса дисциплины.

Лабораторный практикум включает практические задания по двум разделам дисциплины: «Реализация ООП в языке Python» и «Реализация ООП в языке C++». Все работы выполняются в соответствии с заданием, выданным преподавателем.

Практический практикум представляет собой знакомство с библиотекой создания графического пользовательского интерфейса.

Самостоятельная работа обучающихся включает в себя проработку материала лекционного курса дисциплины, подготовку и выполнение лабораторных и практических работ, выполнение курсовой работы, а также подготовку к рубежному контролю (для очной формы обучения) и к зачету (экзамену для очной формы).

Для текущего контроля успеваемости по очной форме обучения используется балльно-рейтинговая система контроля оценки академической активности обучающихся. Для получения высокой оценки настоятельно рекомендуется активно участвовать во время обсуждения материала дисциплины на лекционных и лабораторных занятиях, а также тщательно его прорабатывать при самостоятельной работе.

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.	
	Очная форма обучения 3 семестр	Заочная форма обучения
Самостоятельное изучение тем дисциплины:	-	74
Теоретические концепции объектно-ориентированных языков программирования	-	10
Реализация ООП в языке Python	-	32
Реализация ООП в языке C++	-	32
Подготовка к лабораторным работам	8	4
Подготовка к рубежным контролям (по 3 ч. на каждый рубеж)	6	-
Выполнение курсовой работы	-	36
Подготовка к зачету	18	18
Всего:	32	132

Рекомендуемый режим самостоятельной работы (очная форма, 4 семестр)

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
	Очная форма обучения
Подготовка к практическим работам	1
Выполнение курсовой работы	36
Подготовка к экзамену	27
Всего:	64

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ (для очной формы обучения).
2. Отчеты обучающихся по лабораторным занятиям.
3. Банк заданий к рубежным контролям № 1 и № 2 (для очной формы обучения).
4. Курсовая работа.
5. Вопросы к зачету (для очной и заочной формы обучения) и экзамену (для очной формы обучения).

6.2. Система балльно-рейтинговой оценки работы обучающихся по дисциплине

№	Наименование	Содержание					
Очная форма обучения							
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы <i>(доводятся до сведения обучающихся на первом учебном занятии)</i>	Распределение баллов за 3 семестр					
		Вид учебной работы:	Посещение лекций	Выполнение и защита результатов лабораторных работ	Рубежный контроль №1	Рубежный контроль №2	Зачет
		Балльная оценка:	16 x 8 = 8 б	26 x 16 лаб. = 32 б	15 б	15 б	30 б
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	60 и менее баллов – неудовлетворительно; незачтено; 61...73 – удовлетворительно; зачтено; 74...90 – хорошо; 91...100 – отлично.					

3	Критерии допуска к промежуточной аттестации, возможности получения, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается.</p> <p>Для получения зачета без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность.</p> <p>Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине не снижается.</p> <p>За академическую активность в ходе освоения дисциплины, участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30.</p> <p>Основанием для получения дополнительных баллов являются:</p> <ul style="list-style-type: none"> - выполнение дополнительных заданий по дисциплине; дополнительные баллы начисляются преподавателем; - участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	<p>В случае если к промежуточной аттестации набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра.</p> <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>

№	Наименование	Содержание			
Очная форма обучения					
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы <i>(доводятся до сведения обучающихся на первом учебном занятии)</i>	Распределение баллов за 4 семестр			
		Вид учебной работы:	Выполнение и защита результатов практических работ	Рубежный контроль №3	Рубежный контроль №4
	Балльная оценка:	10б x 4 пр. = 40 б	15 б	15 б	30 б
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	60 и менее баллов – неудовлетворительно; незачтено; 61...73 – удовлетворительно; зачтено; 74...90 – хорошо; 91...100 – отлично.			

3	Критерии допуска к промежуточной аттестации, возможности получения, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается.</p> <p>Для получения экзамена без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность.</p> <p>Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине не снижается.</p> <p>За академическую активность в ходе освоения дисциплины, участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30.</p> <p>Основанием для получения дополнительных баллов являются:</p> <ul style="list-style-type: none"> - выполнение дополнительных заданий по дисциплине; дополнительные баллы начисляются преподавателем; - участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	<p>В случае если к промежуточной аттестации набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра.</p> <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>

5

Критерии оценки курсовой работы

Курсовая работа оценивается отдельно. Максимальная сумма баллов - 100. При оценке качества выполнения работы и уровня защиты используется следующее распределение баллов:

а) Качество пояснительной записки — 40 баллов

б) Качество доклада - 20 баллов.

в) Качество защиты работы — 40 баллов.

При рассмотрении качества пояснительной записки и графической части работы принимается к сведению ритмичность выполнения работы, отсутствие ошибок, логичность и последовательность построения материала, правильность выполнения, соблюдения требований к оформлению работы.

При оценке качества доклада учитывается уровень владения материалом, степень аргументированности, четкости, последовательности и правильности изложения материала, а также соблюдение регламентов.

При оценке уровня качества ответов на вопросы принимается во внимание правильность, полнота и степень ориентированности в материале.

Комиссия по приему защиты курсовой работы оценивает вышеуказанные составляющие компоненты и определяет итоговую оценку.

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в форме письменного тестирования, экзамен и зачет в виде ответа на вопросы.

Перед проведением рубежного контроля преподаватель прорабатывает с обучающимся основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии.

Варианты заданий для рубежных контролей № 1, № 2 состоят из 15 вопросов, для контролей № 3, № 4 — из 15 вопросов. Каждый правильный ответ оценивается в 1 балл.

На каждую подготовку к рубежному контролю обучающемуся отводится академический час.

Преподаватель оценивает в баллах результаты рубежных контролей каждого обучающегося по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

На зачете и экзамене (для очной формы обучения) обучающимся предлагается ответить на 2 вопроса. Вопросы к зачету и экзамену доводятся до обучающихся на последней занятии в семестре. Каждый вопрос оценивается в 15 баллов. На подготовку ответа отводится 1 астрономический час.

Результаты текущего контроля успеваемости, зачета и экзамена заносятся преподавателем в зачетно-экзаменационную ведомость, которая сдается в организационный отдел института в день проведения зачета и экзамена, а также выставляются в зачетную книжку обучающегося.

6.4. Примеры оценочных средств для рубежных контролей, зачета и экзамена

6.4.1 Примеры заданий для рубежного контроля №1

1. Как много конструкторов в классе может иметь Python?
 - а) 0, в них нет необходимости.
 - б) 1, в Python можно создать лишь один конструктор.
 - в) 2, почему бы и нет?!
 - г) Бесконечно много.

2. Что будет выведено на экран?

```
test.py
1 class Test:
2     def print_text(self):
3         print("Это родительский класс Test")
4 class Test2(Test):
5     def print_text(self):
6         print("Это дочерний класс Test2")
7 test2 = Test2()
8 test2.print_text()
9
10
11
12
```

- а) Это родительский класс Test
- б) Это дочерний класс Test2
- в) SyntaxError: invalid syntax
- г) Это родительский класс Test Это дочерний класс Test2

3. Укажите результат выполнения скрипта:

```
class Foo (object):
    obj=0
    def __new__(cls,*dt,**mp):
        return object.__new__(cls,*dt,**mp).obj
    def __add__(self, x):
        return self.obj+2*x
o = Foo()
o+=1
print(o)
```

- 1) 0
- 2) 1
- 3) 2
- 4) 3

6.4.2 Примеры заданий для рубежного контроля №2

1. Класс - это:

- а) любой тип данных, определяемый пользователем;

- б) тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки;
- в) структура, для которой в программе имеются функции работы с нею.

2. Что называется конструктором?

- а) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- б) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- в) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- г) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

3. Выберите правильное объявление производного класса

- а) `class MoreDetails:: Details;`
- б) `class MoreDetails: public class Details;`
- в) `class MoreDetails: public Details;`
- г) `class MoreDetails: class(Details);`

6.4.3 Примеры заданий для рубежного контроля №3

1. Qt - это ...

- а) специальный компилятор языка C++.
- б) интегрированная среда разработки.
- в) программная библиотека для создания GUI.

2. Каким образом можно создать сигналы и слоты?

- а) Только в автоматическом режиме.
- б) Только в ручную все в коде.
- в) Их можно создавать обеими способами.

3. Какой класс предназначен для создания кнопок

- а) `QPushButton;`
- б) `QPushButon;`
- в) `QPressBtn;`
- г) `QButtonPress.`

6.4.4 Примеры заданий для рубежного контроля №4

1. Благодаря чему отслеживаются действия?
 - а) библиотеке Signals;
 - б) классу Events;
 - в) сигналам и слотам.
2. Какой класс ответствен за отображение таблицы?
 - а) QTableWidgetItem;
 - б) QTableWidgetItem;
 - в) QTableWidgetItem.
3. Какой класс ответствен за отображение деревьев?
 - а) QTreeWidgetItem;
 - б) QTreeWidgetItem;
 - в) QTreeWidgetItem.

6.4.5 Примерный перечень вопросов для зачета (очная и заочная форма обучения)

1. Объектная модель. Основные положения объектной модели. Определение объектно-ориентированных языков программирования.
2. Составные части объектного подхода. Абстракция и Инкапсуляция.
3. Составные части объектного подхода. Модульность и иерархия.
4. Понятие объекта. Состояние объекта. Поведение объекта. Виды операций.
5. Отношения между объектами. Роли объектов.
6. Понятие «Класс». Интерфейс и реализация.
7. Отношения между классами.
8. Классы в C++. Описание класс. И спецификаторы доступа к членам класса. Свойства и методы.
9. C++. Конструкторы и деструкторы класса.
10. C++. Определение методов вне класса.
11. C++. Перегрузка операторов.
12. C++. Преобразование типов.
13. C++. Константные методы класса.
14. C++. Статические свойства и методы класса.
15. C++. Дружественность.
16. C++. Наследование. Базовые и производные класса.
17. C++. Доступ к членам базового класса.
18. C++. Перегрузка функций.
19. C++. Иерархия классов.
20. C++. Спецификаторы доступа при наследовании.

21. C++. Виртуальные методы и абстрактные классы.
22. C++. Множественное наследование. Ромбовидное наследование.
23. C++. Исключения.
24. C++. Шаблонные функции и классы.
25. Python. Определение класса. Атрибуты класса. Создание экземпляров класса.
26. Python. Конструктор и деструкторы. Алгоритм поиска атрибутов.
27. Python. Атрибуты класса. Методы класса и статические методы.
28. Python. Абстрактные методы.
29. Python. Магические методы (операторы).
30. Python. Наследование классов.
31. Python. Переопределение методов.
32. Python. Множественное наследование. Миксины.
33. Python. Исключения.

6.4.6 Примерный перечень вопросов для экзамену (для очной формы обучения)

1. Обзор иерархии классов: Qt, QtCore, QtGui, QtWidgets, QtNetwork, QtXml, QSql, QtMultimedia, QtSvg.
2. Механизм сигналов и слотов.
3. Управление автоматически размещением элементов.
4. Элементы отображения: надписи, индикаторы.
5. Кнопки, флажки, переключатели.
6. Элементы ввода: однострочное поле, редакторы текста, счетчики, элементы ввода даты и времени.
7. Элементы выбора: простой список, иерархические списки, таблицы, вкладки.
8. Модель-представление
9. События в Qt.
10. Графика и звук в Qt.

6.7. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

1. Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд. : Пер. с англ. - ООО «И. Д. Вильямс», 2008. - 720 с.

2. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.

3. Лафоре Р. Объектно-ориентированное программирование в C++. СПб.: Издательский дом "Питер", 2013. - 926 с.

4. Шлее М. Qt 5.10. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2018. — 1072 с.

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Техническое обеспечение

№	Наименование	Использование
1	Комплект: ноутбук, медиа-проектор, экран	Для демонстрации иллюстративного материала при чтении лекций.
2	Персональный компьютер стандартной комплектации	Используется в качестве инструмента и объекта исследования при выполнении лабораторных и контрольных работ.

8.2 Программное обеспечение

№	Наименование	Использование
1	g++	Компилятор языка C++
2	Python 3.8+	Интерпретатор языка Python
3	Qt	Библиотека создания GUI

9 РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Сайт дистанционного обучения в ноу (национальный открытый университет) «ИНТУИТ» содержит бесплатные курсы, программы повышения квалификации и профессиональной переподготовки, интересные доклады и другую полезную информацию <http://www.intuit.ru>.

2. Федеральный портал «Российское образование» <http://edu.ru>

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

1. ЭБС «Лань»

2. ЭБС «Консультант студента»

3. ЭБС «Znanium.com»

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение по реализации дисциплины осуществляется с требованиями ФГОС ВО по данной образовательной программе.

12 ДЛЯ СТУДЕНТОВ, ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме онлайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2, либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнений ведущего преподавателя и доводится до сведения обучающихся.

Аннотация
рабочей программы учебной дисциплины

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ
образовательной программы высшего образования –
программы бакалавриата

09.03.04 Программная инженерия

направленность

Программное обеспечение автоматизированных систем

формы обучения – очная и заочная

Трудоемкость освоения дисциплины – 4 зач. ед. (144 акад. часов)

Семестры: 3-й и 4-й для очной и 4-й для заочной формы обучения

Промежуточная аттестация: зачет в 3-м семестре и экзамена в 4-м семестре для очной и зачет в 4-м семестре для заочной формы обучения

Содержание дисциплины

Раздел 1. Теоретические концепции объектно-ориентированных языков программирования

Раздел 2. Реализация ООП в языке Python

Раздел 3. Реализация ООП в языке C++

Раздел 4. Разработка приложений с графическим пользовательским интерфейсом