

**КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)
АВТОНОМНОЙ НЕКОММЕРЧЕСКОЙ ОБРАЗОВАТЕЛЬНОЙ
ОРГАНИЗАЦИИ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Специальность: 10.05.01 Компьютерная безопасность

Специализация: «Разработка защищенного программного обеспечения»

Формы обучения: очная

Объем дисциплины:

в зачетных единицах: 3 з.е.

в академических часах: 108 ак.ч.

Форма промежуточной аттестации: зачет

Рабочая программа по дисциплине «Объектно-ориентированное программирование» по специальности 10.05.01 Компьютерная безопасность, специализации «Разработка защищенного программного обеспечения», составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования – специалитет по специальности 10.05.01 Компьютерная безопасность, утвержденного приказом Министерства науки и высшего образования Российской Федерации от «26» ноября 2020 г. № 1459, профессионального стандарта 06.032 «Специалист по безопасности компьютерных систем и сетей», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 14.09.2022 г. № 533н, профессионального стандарта 06.033 «Специалист по защите информации в автоматизированных системах», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 14.09.2022 г. №525н.

Рабочая программа:

обсуждена и рекомендована к утверждению решением кафедры естественных дисциплин, сервиса и туризма от 25 марта 2025 г., протокол № 6

одобрена Научно-методическим советом Казанского кооперативного института (филиала) от 2 апреля 2025 г., протокол № 3

СОДЕРЖАНИЕ

1. Цель и задачи освоения дисциплины	4
2. Место дисциплины в структуре образовательной программы	4
3. Перечень планируемых результатов обучения по дисциплине	5
4. Объем дисциплины и виды учебной работы.....	6
5. Содержание дисциплины, структурированное по темам (разделам) с указанием количества академических часов и видов учебных занятий.....	7
5.1. Содержание дисциплины	7
5.2. Разделы, темы дисциплины и виды занятий	8
6. Практические занятия	9
7. Лабораторные работы	9
8. Примерная тематика курсовых работ (проектов)	10
9. Самостоятельная работа студента	10
10. Оценивание результатов обучения и уровня сформированности компетенций	16
11. Учебно-методическое обеспечение дисциплины	16
12. Перечень программного обеспечения, профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины.....	17
13. Материально–техническое обеспечение дисциплины	18
Обновление рабочей программы дисциплины (модуля)	20

1. Цель и задачи освоения дисциплины

Цель дисциплины – изучение студентами объектно-ориентированного подхода в программировании. Все практические примеры рассматриваются на языке программирования Java. Воспитательная цель: формирование свободного и творческого подхода к программированию на современных языках высокого уровня, интереса к наблюдению за тенденциями и новостями в области средств разработки программного обеспечения.

Отбор материала основывается на необходимости ознакомить студентов со следующей современной научной информацией:

- об особенностях и последних достижениях в области объектно-ориентированного программирования;
- о положительных и отрицательных чертах подхода к программированию, реализованному в языке Java.

Содержательное наполнение дисциплины обусловлено общими задачами в подготовке бакалавра.

Научной основой для построения программы данной дисциплины является теоретико-прагматический подход в обучении

Основные задачи курса на основе системного подхода:

- закрепление навыков алгоритмизации и программирования на основе углубленного изучения объектно-ориентированного подхода в программировании;
- знакомство с принципами инкапсуляции, наследования и полиморфизма;
- обучение созданию мультиплатформенных приложений

2. Место дисциплины в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» основной образовательной программы – программы специалитета по специальности 10.05.01 Компьютерная безопасность специализация «Разработка защищенного программного обеспечения».

Код компетенции	Предшествующие дисциплины (модули), практики	Изучаемые в текущем семестре дисциплины (модули), практики	Последующие дисциплины (модули), практики
-----------------	--	--	---

ОПК-3 ОПК-7	Теория графов и ее приложения Методы программирования	Объектно-ориентированное программирование Комбинаторный анализ Дифференциальные уравнения	Дискретное программирование Теория игр Теория параллельных алгоритмов Математический анализ Алгебра и аналитическая геометрия Методы оптимизации Практикум по численным методам
----------------	--	--	---

3. Перечень планируемых результатов обучения по дисциплине

Изучение дисциплины направлено на формирование у обучающихся компетенций.

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения
ОПК-3 Способен на основании совокупности математических методов разрабатывать, обосновывать и реализовывать процедуры решения задач профессиональной деятельности	ОПК-3.1 Применяет знания основных математических методов и их модификаций для обоснования процедур решения задач профессиональной деятельности	Знать: цели и задачи проводимых исследований и разработок, методы адаптации существующих математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач. Отечественный и международный опыт в соответствующей области исследований, методы адаптации, существующих математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения, использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач. Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов, использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач Владеть: разработка с использованием и адаптацией существующих математических методов, изменение и согласование архитектуры программного обеспечения

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения
		с системным аналитиком и архитектором программного обеспечения. Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач, использование и адаптирование существующих математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач
ОПК-7 Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7.4 Демонстрирует умение создавать программы на языках высокого уровня, применять методы визуального объектного моделирования в области разработки программ и соответствующие инструментальные средства в процессе решения профессиональных задач	Знать: методы и средства проектирования программного обеспечения при реализации математически сложных алгоритмов, современный отечественный и зарубежный опыт в разработке алгоритмов компьютерной математики, отечественный и международный опыт в разработке современных алгоритмов компьютерной математики Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения эффективно реализующих математически сложные алгоритмы, ориентироваться в современных алгоритмах компьютерной математики, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов Владеть: разработка математически сложных алгоритмов, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения

4. Объем дисциплины и виды учебной работы

Объем дисциплины и виды учебной работы в академических часах с выделением объема контактной работы обучающихся с преподавателем и самостоятельной работы обучающихся

очная форма обучения

Вид учебной деятельности	ак. часов	
	Всего	По семестрам 3
1. Контактная работа обучающихся с преподавателем:	69	69
Аудиторные занятия, часов всего, в том числе:	68	68
• занятия лекционного типа	34	34
• занятия семинарского типа:	34	34

практические занятия		-	-
лабораторные занятия		34	34
в том числе занятия в форме практической подготовки		-	-
Консультации		0,5	0,5
Контактные часы на аттестацию в период экзаменационных сессий		0,5	0,5
2. Самостоятельная работа студентов всего, в том числе		39	39
- подготовка курсовой работы		-	-
- проработка учебного (теоретического) материала		20	20
- выполнение домашних заданий по практическим занятиям		19	19
- подготовка к зачету		-	-
Промежуточная аттестация: зачет		-	-
ИТОГО:	часов	108	108
	зач. ед.	3	3

5. Содержание дисциплины, структурированное по темам (разделам) с указанием количества академических часов и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Основные конструкции ЯП Java.

Примитивные типы данных, основные операции, основные операторы, функция main

Тема 2. Основы ООП

Понятие класса, три базовых свойства ООП: инкапсуляция, наследование, полиморфизм, конструкторы, создание экземпляров объектного типа

Тема 3. Методы. Модификаторы доступа.

Структура объявления методов. Совмещение методов. Модификаторы доступа. Статические методы.

Тема 4. Наследование. Глобальный суперкласс Object.

Наследование. Ключевое слово super. Замещение методов. Использование final методов. Глобальный суперкласс Object и его методы equals, hashCode, clone, getClass, toString

Тема 5. Полиморфизм.

Свойство полиморфизма. Динамическое назначение методов.

Тема 6. Абстрактные классы

Абстрактные классы и абстрактные методы. Ограничения при использовании абстрактных классов. Внутренние классы

Тема 7. Интерфейсы

Интерфейсы. Реализация интерфейсов классом. Наследование интерфейсов.

Тема 8. Исключения.

Иерархия классов-исключений в Java. Блоки try, catch, finally. Множественный catch. Ручная генерация исключений (throw), ключевое слово throws. Создание собственных классов исключений.

Тема 9. Дженерики

Классы-оболочки. Параметризованные типы. Параметризованные классы. Параметризованные методы. Параметризованные интерфейсы. Ограничения при использовании дженериков.

Тема 10. Коллекции.

Иерархия интерфейсов коллекций в Java. Интерфейс List и классы, его реализующие, интерфейс Set и классы, его реализующие, интерфейс Queue и классы, его реализующие, интерфейс Map и классы, его реализующие

Тема 11. Лямбда-выражения

Лямбда-выражение, лямбда-оператор, функциональный интерфейс, передача параметров в лямбда-выражение, обобщенный функциональный интерфейс

Тема 12. Функциональные интерфейсы

Функциональные интерфейсы Predicate, Consumer, Function, Supplier, UnaryOperator, BinaryOperator

Тема 13. Основы создания оконных приложений

Пакеты AWT и Swing, компоненты JFrame, JPanel, JButton, JTextField, JTextArea, добавление компонентов на фрейм, позиционирование компонентов основные методы класса Component

Тема 14. Обработка событий

Общая схема обработки событий в Java, интерфейсы-слушатели: ActionListener, WindowListener, KeyListener, MouseListener, классы-адаптеры

5.2. Разделы, темы дисциплины и виды занятий

очная форма обучения

№ п/п	Наименование раздела, темы дисциплины	Виды занятий, включая самостоятельную работу студентов (в часах)	Индикаторы достижения
-------	---------------------------------------	--	-----------------------

		занятия лекционного типа	занятия семинарского типа / из них в форме практической подготовки (лабораторные работы)	самос- тоятельная работа	компетенций
1	Основные конструкции ЯП Java.	2	2/2	2	ОПК-3.1, ОПК-7.4
2	Основы ООП	2	2/2	2	ОПК-3.1, ОПК-7.4
3	Методы. Модификаторы доступа	2	2/2	2	ОПК-3.1, ОПК-7.4
4	Наследование. Глобальный суперкласс Object.	2	2/2	2	ОПК-3.1, ОПК-7.4
5	Полиморфизм.	2	2/2	2	ОПК-3.1, ОПК-7.4
6	Абстрактные классы.	2	2/2	2	ОПК-3.1, ОПК-7.4
7	Интерфейсы	2	2/2	2	ОПК-3.1, ОПК-7.4
8	Исключения.	2	2/2	2	ОПК-3.1, ОПК-7.4
9	Дженерики.	2	2/2	2	ОПК-3.1, ОПК-7.4
10	Коллекции.	2	2/2	2	ОПК-3.1, ОПК-7.4
11	Лямбда-выражения.	2	2/2	2	ОПК-3.1, ОПК-7.4
12	Функциональные интерфейсы	4	4/4	4	ОПК-3.1, ОПК-7.4
13	Основы создания оконных приложений	4	4/4	4	ОПК-3.1, ОПК-7.4
14	Обработка событий.	4	4/4	9	ОПК-3.1, ОПК-7.4
	Всего	34	34/34	39	

6. Практические занятия

Практические занятия не предусмотрены.

7. Лабораторные работы

№ п/п	Наименование раздела, темы дисциплины	Содержание лабораторных работ	Объем (час.)
			Очная форма обучения
1.	Основные конструкции ЯП Java	Примитивные типы данных, основные операции, основные операторы, функция main	2
2.	Основы ООП.	Понятие класса, три базовых свойства ООП: инкапсуляция, наследование, полиморфизм,	2

№ п/п	Наименование раздела, темы дисциплины	Содержание лабораторных работ	Объем (час.)
			Очная форма обучения
		конструкторы, создание экземпляров объектного типа	
3.	Методы. Модификаторы доступа	Структура объявления методов. Совмещение методов. Модификаторы доступа. Статические методы.	2
4.	Наследование. Глобальный суперкласс Object.	Наследование. Ключевое слово super. Замещение методов. Использование final методов. Глобальный суперкласс Object и его методы equals, hashCode, clone, getClass, toString	2
5.	Полиморфизм.	Свойство полиморфизма. Динамическое назначение методов.	2
6.	Абстрактные классы.	Абстрактные классы и абстрактные методы. Ограничения при использовании абстрактных классов. Внутренние классы.	2
7.	Интерфейсы.	Интерфейсы. Реализация интерфейсов классом. Наследование интерфейсов..	2
8.	Исключения	Иерархия классов-исключений в Java. Блоки try, catch, finally. Множественный catch. Ручная генерация исключений (throw), ключевое слово throws. Создание собственных классов исключений.	2
9.	Дженерики.	Классы-оболочки. Параметризованные типы. Параметризованные классы. Параметризованные методы. Параметризованные интерфейсы. Ограничения при использовании дженериков.	2
10.	Коллекции	Иерархия интерфейсов коллекций в Java. Интерфейс List и классы, его реализующие, интерфейс Set и классы, его реализующие, интерфейс Queue и классы, его реализующие, интерфейс Map и классы, его реализующие	2
11.	Лямбда- выражения.	Лямбда-выражение, лямбда-оператор, функциональный интерфейс, передача параметров в лямбда-выражение, обобщенный функциональный интерфейс	2
12.	Функциональные интерфейсы	Функциональные интерфейсы Predicate, Consumer, Function, Supplier, UnaryOperator, BinaryOperator	4
13.	Основы создания оконных приложений.	Пакеты AWT и Swing, компоненты JFrame, JPanel, JButton, JTextField, JTextArea, добавление компонентов на фрейм, позиционирование компонентов основные методы класса Component	4
14.	Обработка событий	Общая схема обработки событий в Java, интерфейсы слушатели: ActionListener, WindowListener, KeyListener, MouseListener, классы-адаптеры	4
	Всего		34

8. Примерная тематика курсовых работ (проектов)

Курсовые работы (проекты) не предусмотрены.

9. Самостоятельная работа студента

Самостоятельная работа студента при изучении дисциплины «Объектно-ориентированное программирование» направлена на:

- освоение нормативных правовых актов и учебной литературы, необходимых для освоения дисциплины;
- самостоятельный поиск информации в Интернете и других источниках;
- выполнение домашних заданий по практическим занятиям;
- подготовку к зачету.

Краткие рекомендации по выполнению самостоятельной работы:

Успешное усвоение дисциплины предполагает большой, упорный, серьезный, систематический труд студентов. Важнейшая его составная часть – выполнение разных видов самостоятельной работы.

1. Составление тематического конспекта на основе изученной основной и дополнительной учебной литературы. В тематическом конспекте за основу берется содержание темы, вопросы для обсуждения.

Этапы работы.

1.1 Конспектирование делается только после того, как прочитан или усвоен материал для конспектирования.

1.2. Необходимо мысленно или письменно составить план конспекта. По этому плану и будет строиться конспект далее.

1.3. Составление самого конспекта. Можно сказать, что конспект – это расширенные тезисы, дополненные рассуждениями и доказательствами, содержащимися в материалах для конспекта, а также собственными мыслями и положениями составителя конспекта.

Писать конспект рекомендуется четко и разборчиво. В конспекте можно выделять места текста в зависимости от их значимости. Для этого применяются различного размера буквы, подчеркивания, замечания на полях.

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Определить класс `train`, содержащий поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа `train`, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массив по пункту назначения.

2. Создать абстрактный класс Телефонный справочник с методами, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска. Создать производные классы: Персона (фамилия, адрес, номер телефона), Организация (название, адрес, телефон, факс, контактное лицо), Друг (фамилия, адрес, номер телефона, дата рождения) со своими методами вывода информации на экран, и определения соответствия искомому типу. Создать массив из n элементов различных классов, вывести полную

информацию из массива на экран, а также организовать поиск в массиве по фамилии.

3. Решить с использованием формы: определить класс для работы с комплексными числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

4. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям.

5. Создать базовый класс – «работник» и производные классы – «служащий с почасовой оплатой», «служащий в штате» и «служащий с процентной ставкой». Определить в каждом классе функцию начисления зарплаты.

6. Создать абстрактный класс Фигура с функцией – площадь и периметр. Реализовать интерфейс в классах: «круг», «прямоугольный треугольник», «трапеция» со своими функциями площади и периметра. Для проверки определить массив ссылок на интерфейс, которым присваиваются адреса различных объектов.

7. Определить интерфейс Тело с функциями вычисления объема и площади поверхности. Реализовать этот интерфейс в классах Конус и Пирамида. Для проверки определить массив ссылок на интерфейс, которым присваиваются адреса различных объектов

8. Определить параметризованный класс Вектор с методами добавления элемента, удаления заданного элемента, замены порядка элементов на обратный. Создать экземпляры этого класса для параметра типа Double и String

Типовые задания для самостоятельной работы

1. Описать класс параллелепипед. Предусмотреть методы создания объектов, перемещения, изменения размеров, вычисления высоты, нахождения центра тяжести. Написать программу, демонстрирующую работу с этим классом.

2. Описать класс «телефонный справочник». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку, добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом.

3. Описать класс «список». Предусмотреть возможность добавления и удаления элементов в начало, конец списка, в указанную позицию, сортировку по убыванию и возрастанию, подсчет количества элементов в списке, вывод на экран части списка с указанием границ вывода.

4. Определить класс «Автомобиль», экземпляры которого имеют поля – марка, мощность, объем двигателя. Определить функции назначения и изменения полей и функцию вывода на экран. Определить производный класс – такси, экземпляры которого имеют поле – тариф. Определить

функцию вывода на экран. Определить функцию переназначения тарифа и функцию расчета стоимости поездки.

5. Определить класс «Жидкость», экземпляры которого имеют поля – название, плотность. Определить функции назначения и изменения полей и функцию вывода на экран. Определить производный класс – раствор кислоты, экземпляры которого имеют поле – концентрация. Определить функцию вывода на экран. Определить функцию переназначения концентрации.

6. Определить класс «Автомобиль», экземпляры которого имеют поля – марка, мощность, объем двигателя. Определить функции назначения и изменения полей и функцию вывода на экран. Определить производный класс – грузовой автомобиль, экземпляры которого имеют поле – грузоподъемность. Определить функцию вывода на экран. Определить функцию переназначения грузоподъемности и функцию, определяющую возможность перевозки груза.

7. Определить класс «Работник», и производные классы – «Служащий с почасовой оплатой», «Служащий в штате», «Служащий с процентной ставкой». Определить функции назначения и изменения полей и функцию вывода на экран и функцию назначения заработной платы.

8. Определить интерфейс Root с методами вычисления вычисления корней и вывода результата на экран. Реализовать интерфейс в классах Linear и Square.

9. Определить интерфейс Function с методами вычисления функции в заданной точке и вывода результата на экран. Реализовать интерфейс в классах Ellipse и Hyperbola.

10. Определить интерфейс Pair с методами арифметическими операциями. Реализовать интерфейс в классах Complex и Rational (рациональная дробь).

11. Определить интерфейс Triad с методами увеличения на 1. Реализовать интерфейс в классах Date и Time.

12. Определить интерфейс Pair с методами арифметическими операциями. Реализовать интерфейс в классах Money и Fraction (дробное число).

13. Нарисовать ромб, заданный стороной и двумя углами, треугольник, заданный координатами вершин, и эллипс. Выделить область пересечения этих фигур.

14. Нарисовать ромб, заданный стороной и двумя углами, прямоугольник, заданный координатами концов своей диагонали, и окружность, заданную координатами центра и радиусом. Выделить область пересечения этих фигур.

15. Нарисовать произвольный восьмиугольник, треугольник, заданный координатами своих вершин, и эллипс. Выделить область пересечения этих фигур.16

16. Нарисовать два треугольника, заданных координатами своих вершин. В треугольнике с большей площадью нарисовать медиану меньшего

угла. В треугольнике с меньшей площадью нарисовать биссектрисы всех углов.

17. Написать программу, демонстрирующую работу с объектами двух типов, T1 и T2, для чего создать систему соответствующих классов. Каждый объект должен иметь идентификатор (в виде произвольной строки символов) и одно или несколько полей для хранения состояния (текущего значения) объекта.

Перечень типов объектов: Треугольник (Triangle), Квадрат (Quad), Прямоугольник (Rectangle), Четырехугольник (Tetragon), Пятиугольник (Pentagon).

Перечень дополнительных методов:

- move() переместить объект на плоскости
- compare (T1 t1, T2 t2) сравнение объектов по площади
- isIntersect(T1 t1, T2 t2) определяет факт пересечения объектов
- isInclude(T1 t1, T2 t2) определяет факт включения t2 в t1

Вариант	T1	T2	Методы
1	Triangle	Quad	Move, compare
2	Quad	Pentagon	Move, isIntersect
3	Triangle	Rectangle	Move, compare

18. Написать программу для моделирования Т-образного сортировочного узла на железной дороге с использованием класса, реализующего интерфейс Deque. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность ввода исходных данных с клавиатуры и из файла.

19. Написать программу, моделирующую управление каталогом в файловой системе. Для каждого файла в каталоге содержатся следующие сведения: имя файла, дата создания, количество обращений к файлу. Программа должна обеспечивать:

- начальное формирование каталога файлов;
- вывод каталога файлов;
- удаление файлов, дата создания которых раньше заданной;
- выборку файла с наибольшим количеством обращений.

Выбор конкретного действия должен осуществляться с помощью меню. Для представления каталога использовать класс, реализующий интерфейс List.

20. Написать программу моделирования работы автобусного парка. Сведения о каждом автобусе содержат: номер автобуса, фамилию и инициалы водителя, номер маршрута. Обеспечить выбор с помощью меню и выполнение следующих функций:

- начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла);
- имитация выезда автобуса из парка: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов,

находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;

- имитация въезда автобуса в парк: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;

- вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте.

Для представления необходимых списков использовать класс, реализующий интерфейс List.¹⁷

21. Написать программу учета заявок на авиабилеты. Каждая заявка содержит: пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета. Обеспечить выбор с помощью меню и выполнение следующих функций:

- добавление заявок в список;
- удаление заявок;
- вывод заявок по заданному номеру рейса и дате вылета;
- вывод всех заявок.

Для хранения данных использовать класс, реализующий интерфейс List.

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

Типовые задачи к зачету

1. Создать класс точка, предусмотреть в нем несколько конструкторов, методы определения координатной четверти, которой принадлежит точка, перемещения точки, метод проверяющий являются ли две точки симметричными относительно начала координат, метод, вычисляющий расстояние между двумя точками.

2. Создать класс Равнобедренная трапеция, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура равнобедренной трапецией; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N трапеций, найти количество трапеций, у которых площадь больше средней площади.

3. Создать класс Angle для работы с углами на плоскости, задаваемые в градусах и минутах. Обязательно должны быть реализованы перевод в радианы, приведение к диапазону 0-360, увеличение и уменьшение угла на заданную величину, получение синуса, сравнение углов.

4. Создать класс Date для работы с датами в формате «год.месяц.день». Дата представляется структурой с тремя полями. Класс должен включать не менее трех методов инициализации: числами, строкой и датой. Обязательными операциями являются: вычисление даты через заданное количество дней, вычитание заданного количества дней из даты, определение високосности года, присвоение и получение отдельных частей, сравнение дат, вычисление количества дней между датами.

5. Создать абстрактный класс Figure с методами вычисления площади и периметра. Создать производные классы Rectangle (прямоугольник), Circle (круг), Trapezium(трапеция) со своими методами вычисления площади и периметра. Самостоятельно определить какие поля можно задать в базовом классе, а какие - в производных.

6. Определить класс «Работник», и производные классы – «Служащий с почасовой оплатой», «Служащий в штате», «Служащий с процентной ставкой». Определить функции назначения и изменения полей и функцию вывода на экран и функцию назначения заработной платы.

7. Определить интерфейс Root с методами вычисления вычисления корней и вывода результата на экран. Реализовать интерфейс в классах Linear и Square.

8. Определить интерфейс Function с методами вычисления функции в заданной точке и вывода результата на экран. Реализовать интерфейс в классах Ellipse и Hyperbola.

9. Определить интерфейс Pair с методами арифметическими операциями. Реализовать интерфейс в классах Complex и Rational (рациональная дробь).

10. Определить интерфейс Triad с методами увеличения на 1. Реализовать интерфейс в классах Date и Time. 18

11. Определить интерфейс Pair с методами арифметическими операциями. Реализовать интерфейс в классах Money и Fraction (дробное число).

12. Нарисовать ромб, заданный стороной и двумя углами, треугольник, заданный координатами вершин, и эллипс. Выделить область пересечения этих фигур.

10. Оценивание результатов обучения и уровня сформированности компетенций

Оценивание результатов обучения по дисциплине и уровня сформированности компетенций (части компетенции) осуществляется в рамках текущего контроля успеваемости и промежуточной аттестации в соответствии с оценочными материалами.

11. Учебно-методическое обеспечение дисциплины

Основная литература:

1. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2021. — 206 с. —(Высшее образование). — ISBN 978-5-534-00849-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470223>

2. Бабушкина, И. А. Практикум по объектно-ориентированному программированию : [16+] / И. А. Бабушкина, С. М. Окулов. — 5-е изд.,

электрон. – Москва : Лаборатория знаний, 2020. – 369 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=221691> (дата обращения: 15.08.2021). – Библиогр.: с. 358. – ISBN 978-5-00101-780-6. – Текст : электронный.

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах «Лань» и «Юрайт»

Дополнительная литература:

1. Николаев, Е.И. Объектно-ориентированное программирование: учебное пособие/ Е.И. Николаев; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2015. - 225 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458133>

Электронные библиотечные системы (ЭБС) и электронные образовательные ресурсы

1. Электронно-библиотечная система ZNANIUM <https://www.znanium.com>
2. Универсальная справочно-информационная полнотекстовая база данных периодических изданий «East View» (ИБИС) <https://dlib.eastview.com/>
3. Электронно-библиотечная система «BOOK.ru» www.book.ru
4. Электронно-библиотечная система «IPRbooks» <http://www.iprbookshop.ru/>
5. ЭБС «Национальный цифровой ресурс «Руконт» <https://lib.rucont.ru/search>
6. Образовательная платформа ЮРАЙТ <http://www.urait.ru>

Электронный каталог Университета – Автоматизированная информационная библиотечная система (АИБС «МегаПро»)

12. Перечень программного обеспечения, профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины

Лицензионное программное обеспечение, в том числе отечественного производства:

– DsktpEdu ALNG LicSARk OLV F1 1Y Acdmc Ent (MS Windows) (подписка на ПО).

Свободно распространяемое программное обеспечение, в том числе отечественного производства:

– 7-Zip (свободный файловый архиватор с высокой степенью сжатия

данных) (отечественное ПО).

- FastStone Image Viewer (программа для просмотра изображений в Microsoft Windows (лицензия бесплатного программного обеспечения)).

- Foxit Reader (Бесплатное прикладное программное обеспечение для просмотра электронных документов в стандарте PDF (лицензия бесплатного программного обеспечения)).

- Indigo (система программ для создания и проведения компьютерного тестирования знаний).

- Google Chrome, ЯндексБраузер (браузер).

- Adobe Acrobat Reader DC (• ПО для просмотра, печати и комментирования документов в формате PDF)

- Visual Studio Code (интегрированная среда разработки программного обеспечения)

Профессиональные базы данных не используются.

Информационные справочные системы:

- СПС КонсультантПлюс. Компьютерная справочная правовая система, широко используется учеными, студентами и преподавателями (подписка на ПО)

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронно-библиотечной

13. Материально–техническое обеспечение дисциплины

Образовательный процесс обеспечен учебными аудиториями для проведения учебных занятий, а именно для проведения занятий лекционного типа, лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещениями для самостоятельной работы студентов.

Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Рабочее место преподавателя (стол, стул), рабочие места обучающихся, доска для написания мелом, кафедра.

Технические средства обучения: персональный компьютер-1, мультимедийное оборудование (проектор, проекционный экран) - 1

Учебная аудитория для проведения лабораторных занятий. Аудитория информационных технологий

Рабочее место преподавателя (стол, стул), рабочие места обучающихся:

Технические средства обучения:

персональный компьютер с подключением к сети «Интернет» - 15, сетевое оборудование.

Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет»

и обеспечением доступа в электронную информационно-образовательную среду института.

Рабочие места обучающихся.

Технические средства обучения: персональный компьютер - 13.

Учебные аудитории соответствуют действующим противопожарным правилам и нормам, укомплектованы учебной мебелью.

Обновление рабочей программы дисциплины (модуля)

Наименование раздела рабочей программы, в который внесены изменения

(измененное содержание раздела)

обсуждено и рекомендовано к утверждению решением кафедры
наименование кафедры

от ____ ____ 20__ г., протокол № ____

одобрено Научно-методическим советом ____ от ____ ____ 20__ г.,
протокол № ____