



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

***ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО
ПРОГРАММИРОВАНИЯ***

Направление подготовки (специальность)
27.03.04 Управление в технических системах

Направленность (профиль/специализация) программы
Системы и средства автоматизации технологических процессов

Уровень высшего образования - бакалавриат

Форма обучения
заочная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Автоматизированных систем управления
Курс	3

Магнитогорск
2025 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 27.03.04 Управление в технических системах (приказ Минобрнауки России от 31.07.2020 г. № 871)

Рабочая программа рассмотрена и одобрена на заседании кафедры
Автоматизированных систем управления
29.01.2025, протокол № 6

Зав. кафедрой _____ С.М. Андреев

Рабочая программа одобрена методической комиссией ИЭиАС
04.02.2025 г. протокол № 3

Председатель _____ В.Р. Храмшин

Рабочая программа составлена:
доцент кафедры АСУ, канд. техн. наук

_____ Е.С. Рябчикова

Рецензент:
Технический директор ЗАО «КонсОМ СКС»

_____ Е.Ю. Васильев



Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

1 Цели освоения дисциплины (модуля)

- теоретическая и практическая подготовка, обеспечивающая получение знаний по основам объектно-ориентированного программирования;
- получение практических навыков разработки объектно-ориентированных программ;
- получение навыков использования стандартных приемов при составлении и отладке объектно-ориентированных программ на персональных компьютерах;
- получение навыков использования объектно-ориентированного подхода к решению практических задач

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Основы объектно-ориентированного программирования входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Информатика

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Интеллектуальные системы управления

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Основы объектно-ориентированного программирования» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-6	Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности
ОПК-6.1	Разрабатывает алгоритмы и программы для практического применения в сфере своей профессиональной деятельности
ОПК-6.2	Использует современные методы и средства контроля, диагностики и управления для практического применения в сфере своей профессиональной деятельности
ОПК-6.3	Использует современные информационные технологии в сфере своей профессиональной деятельности

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 2 зачетных единиц 72 акад. часов, в том числе:

- контактная работа – 4,4 акад. часов;
 - аудиторная – 4 акад. часов;
 - внеаудиторная – 0,4 акад. часов;
 - самостоятельная работа – 63,7 акад. часов;
 - в форме практической подготовки – 0 акад. час;
 - подготовка к зачёту – 3,9 акад. час
- Форма аттестации - зачет

Раздел/ тема дисциплины	Курс	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основы ООП								
1.1 Основные принципы ООП. Класс и объект. Создание экземпляров класса. Оператор «точка». Переменные-члены и методы-члены класса.	3	1			2	Самостоятельное изучение учебной литературы	Собеседование	ОПК-6.1
1.2 Конструкторы. Модификаторы управления доступа и области видимости. Скрытие информации и инкапсуляция.		1			4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ОПК-6.1
1.3 Геттеры и сеттеры. Ключевое слово «this». Метод «toString». Константы					4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ОПК-6.1
Итого по разделу		2			10			
2. Наследование								
2.1 Области видимости. Переопределение методов и скрытие полей.	3				2	Самостоятельное изучение учебной литературы.	Собеседование	ОПК-6.1
2.2 Ключевое слово «super». Конструктор без параметров по умолчанию.				2	8	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ОПК-6.1

2.3 Одиночное наследование	3				8	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ОПК-6.1
Итого по разделу				2	18			
3. Полиморфизм, абстрактные классы и интерфейсы								
3.1 Подстановка	3				2	Самостоятельное изучение учебной литературы.	Собеседование	ОПК-6.1
3.2 Апкастинг и даункастинг.					6	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ОПК-6.1
3.3 Оператор «instanceof»					10	Самостоятельное изучение учебной литературы.	Собеседование	ОПК-6.1
Итого по разделу					18			
4. Абстрактные классы и интерфейсы								
4.1 Абстрактный метод. Абстрактный класс. Интерфейс	3				6	Самостоятельное изучение учебной литературы.	Собеседование	ОПК-6.1
4.2 Реализация множественных интерфейсов. Интерфейс и абстрактный суперкласс					6	Самостоятельное изучение учебной литературы.	Собеседование	ОПК-6.1
4.3 Динамическое связывание. Инкапсуляция, связывание и связность.					5,7	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ОПК-6.1
Итого по разделу					17,7			
Итого за семестр		2		2	63,7		зачёт	
Итого по дисциплине		2		2	63,7		зачет	

5 Образовательные технологии

Для реализации предусмотренных видов учебной работы в качестве образовательных технологий в преподавании дисциплины «Основы объектно-ориентированного программирования» используются традиционная и модульно-компетентностная технологии.

Передача необходимых теоретических знаний и формирование основных представлений по курсу происходит с использованием мультимедийного оборудования.

Теоретический курс включает: вводную лекцию, первое представление о предмете и знакомит студентов с назначением и задачами курса; проблемные лекции являются результатом усвоения полученной информации посредством постановки проблемного вопроса и поиска путей его решения; лекции – консультации, изложение нового материала сопровождается постановкой вопросов и дискуссией в поисках ответов на эти вопросы.

В ходе проведения лекционных занятий предусматривается:

- использование электронного демонстрационного материала по темам объектно-ориентированного программирования;
- использование электронных учебников по отдельным темам занятий;
- встречи с представителями проектных и обслуживающих предприятий: встречи с представителями проектных и обслуживающих предприятий: ООО «ОСК», ООО «Информсервис ММК», ЗАО «КонсОМ»; предполагаемые темы встреч «Применение технологии объектно-ориентированного программирования для разработки обучающих программ-тренажеров», «Программное обеспечение современной системы управления»;
- активные и интерактивные формы обучения: вариативный опрос, устный опрос, индивидуальная «защита» практических работ и т.д.

Лекционный материал закрепляется в ходе практических работ, на которых выполняются групповые или индивидуальные задания по пройденной теме. Часть практических занятий проводится в интерактивной форме с использованием следующих методов:

- работа в команде, предусматривает совместную деятельность студентов в группе под руководством лидера, направленную на решение общей задачи с делением ответственности и полномочий;
- проблемное обучение, которое заключается в стимулировании студентов к самостоятельной «добыче» знаний, необходимых для решения конкретной проблемы;
- контекстное обучение, которое позволяет усвоить материал путем выявления связей между конкретным знанием и его применением;
- обучение на основе опыта, активизация познавательной деятельности студентов за счет ассоциации их собственного опыта с предметом изучения.

Самостоятельная работа стимулирует студентов к самостоятельной проработке в процессе подготовки к устному опросу и итоговой аттестации.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1020593> (дата обращения: 19.01.2025). – Режим доступа: по подписке.

2. Пруцков, А. В. Язык программирования Java. Введение в курс: объектно-ориентированное программирование : учебное пособие / А. В. Пруцков. — Рязань : РГРТУ, 2016. — 56 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/168308> (дата обращения: 19.01.2025). — Режим доступа: для авториз. пользователей.

б) Дополнительная литература:

1. Барков, И. А. Объектно-ориентированное программирование / И. А. Барков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 700 с. — ISBN 978-5-507-47113-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/329549> (дата обращения: 19.01.2025). — Режим доступа: для авториз. пользователей.

2. Унгер, А. Ю. Объектно-ориентированное программирование : учебник / А. Ю. Унгер. — Москва : РТУ МИРЭА, 2023. — 102 с. — ISBN 978-5-7339-2051-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/398276> (дата обращения: 19.01.2025). — Режим доступа: для авториз. пользователей.

3. Сорокин, А. А. Объектно-ориентированное программирование : учебное пособие / А. А. Сорокин. — Ставрополь : СКФУ, 2014. — 174 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/155255> (дата обращения: 19.01.2025). — Режим доступа: для авториз. пользователей.

4. Объектно-ориентированное программирование на C++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. — Красноярск : СФУ, 2019. — 288 с. — ISBN 978-5-7638-4034-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/157572> (дата обращения: 19.01.2025). — Режим доступа: для авториз. пользователей.

в) Методические указания:

1. Васюткина, И. А. Технология разработки объектно-ориентированных программ на JAVA / Васюткина И.А. - Новосибирск : НГТУ, 2012. - 152 с.: ISBN 978-5-7782-1973-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/557111> (дата обращения: 19.01.2025). – Режим доступа: по подписке.

2. Молчанова, Е. И. Объектно-ориентированное программирование. Основы объектного программирования на языке Java в среде IDE NetBeans : учебное пособие : в 2 частях / Е. И. Молчанова, В. В. Федоров. — Иркутск : ИрГУПС, 2022 — Часть 1 — 2022. — 128 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/342125> (дата обращения: 19.01.2025). — Режим доступа: для авториз. пользователей.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
7Zip	свободно распространяемое ПО	бессрочно
MS Office 2003 Professional	№ 135 от 17.09.2007	бессрочно
FAR Manager	свободно распространяемое ПО	бессрочно

Eclipse	свободно распространяемое ПО	бессрочно
---------	------------------------------	-----------

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Электронная база периодических изданий East View Information Services, ООО «ИВИС»	https://dlib.eastview.com/
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/
Российская Государственная библиотека. Каталоги	https://www.rsl.ru/ru/4readers/catalogues/
Электронные ресурсы библиотеки МГТУ им. Г.И. Носова	https://host.megaprolib.net/M/P0109/Web
Международная реферативная и полнотекстовая справочная база данных научных изданий «Springer Nature»	https://www.nature.com/siteindex

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Учебные аудитории для проведения занятий лекционного типа (ауд. 437)
Мультимедийные средства хранения, передачи и представления информации
2. Учебная аудитория для проведения практических занятий: компьютерный

класс (ауд. 448)

Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета

3. Помещения для самостоятельной работы обучающихся (ауд. 448)

Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета

4. Учебные аудитории для групповых и индивидуальных консультаций, текущего контроля и промежуточных консультаций (ауд. 448)

Доска, мультимедийный проектор, экран

5. Помещение для хранения и профилактического обслуживания учебного оборудования (ауд. 445)

Стеллажи для хранения учебно-методической документации

Учебно-методическое обеспечение самостоятельной работы студентов

По дисциплине «**Основы объектно-ориентированного программирования**» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов на практических занятиях происходит под контролем преподавателя в ходе выполнения практических работ, при решении задач и выполнении упражнений, которые для студентов определяет преподаватель. Внеаудиторная самостоятельная работа осуществляется в виде проработки материала практических занятий, выполнения домашних заданий и при консультациях с преподавателем.

Примерные вопросы для устного опроса по выполненным практическим работам

Тема практической работы	Вопросы для устного опроса
Основы языка Java. Массивы, примитивные типы, объявление классов	1. Какие существуют виды исключительных ситуаций? 2. Какие классы вы использовали для проверки типа элемента последовательности? 3. Почему все методы, реализующие операции, указанные в задании объявляются как статические? 4. Что такое расширяющее преобразование типов? 5. Что такое примитивный тип в Java?
Основы Языка Java. Перегрузка и перекрытие методов, наследование	1. Дайте определения понятиям статический метод и статическое поле класса. 2. Что такое явное и неявное приведение типов? 3. Какие методы классов-оболочек над примитивными типами используются для получения значения примитивного типа из его строкового представления. 4. Для какого примитивного типа не существует класса-оболочки?
Основы Языка Java. Перегрузка и перекрытие методов, наследование. Классы-оболочки	1. Каким образом реализуется наследования в Java? 2. Каким образом класс-потомок может обратиться полям и методам суперкласса? 3. В какой последовательности осуществляется вызов конструкторов классов, являющихся суперклассами для данного класса?
Основы Языка Java. Наследование, тригонометрические функции класса Math	1. Для чего предназначена секция импорта? 2. Какие существуют ограничения и правила именования классов? 3. Какие существуют способы инициализации массивов?
Основы языка Java. Работа с изменяемыми и неизменяемыми строками	1. В чем состоит принципиальное отличие между классами String и StringBuffer? 2. Какие типы исключительных ситуаций вы знаете? К исключительным ситуациям какого типа относится java.lang.Error? 3. Допустимо ли преобразование объекта класса Integer к типу String? 4. Какие методы предусмотрены в классе Integer для представления числовых значений в различных системах счисления?
Основы языка Java. Наследование. Сравнение объектов	1. Для чего предназначена перегрузка методов класса? 2. Могут ли перекрываться статические методы класса в классах потомках? 3. Какие основные задачи решает класс File?

Тема практической работы	Вопросы для устного опроса
	4. Для чего предназначен интерфейс Map?
Основы языка Java. Наследование. Сравнение объектов. Запись в файловый поток	1. Можно ли производить запись в файл с использованием класса OutputStreamWriter? 2. Для чего предназначены буферизованные потоки чтения и записи? 3. Что понимается под блоком инициализации класса? 4. Каким образом функционирует «сборщик мусора» виртуальной машины Java?
ООП в JAVA, наследование, сериализация, файловые потоки ввода-вывода	1. Для чего предназначен интерфейс Serializable? 2. Какие виды наследования поддерживаются в языке Java? 3. Для чего предназначены итераторы? 4. Какие существуют способы задания констант в Java?
ООП в Java. Наследование	1. Какие основные методы класса String вам известны? 2. Какие существуют правила перекрытия методов в Java? 3. Каким образом используются диагностические утверждения?
Обработка изменяемых строк, коллекции, карты	1. Какой метод организации данных использует класс HashMap? 2. Опишите иерархию классов и интерфейсов коллекций в платформе J2EE? 3. Для чего предназначены цепочки конструкторов? 4. Какие методы существуют для получения значения системного времени в J2SE?
Коллекции и списки	1. Для чего предназначены интерфейсы Comparator и Comparable? Реализует ли интерфейс Comparable класс Boolean? 2. Для чего предназначены неизменяемые оболочки коллекций? 3. Какие существуют способы инициализации массивов? 4. Для чего предназначен интерфейс Map?
Наследование. Стандартные потоки ввода-вывода	1. Какой класс удобнее всего использовать для хранения пар типа «ключ-значение»? 2. Существуют ли методы позволяющие сохранять текст в файл с выбранной кодировкой? 3. Какие кодировки поддерживаются для объектов класса String? 4. Что подразумевает понятие рефлексии в Java? 5. Для чего предназначен модификатор transient? Для каких целей предназначен метод sleep у потока?
Файлы. Файловые потоки ввода-вывода. Операции с файлами	1. Какие существуют классы расширяющие класс InputStream и каково их назначение? 2. Какие классы предназначены для преобразования из байтового потока вывода в символьный поток вывода? 3. Каким образом можно получить данные о размере файла? 4. На основе какой структуры данных реализован класс TreeSet? 5. Для каких целей используется модификатор final в объявлении классов? 6. Какова область видимости у классов без явно указанного модификатора доступа?
Многопоточные приложения. Синхронизация	1. Какая последовательность вызова методов классов пакета java.io. при осуществлении чтения из файлового потока? 2. Объясните понятие и назначение финализации объектов. 3. Приведите примеры использования класса ArrayList. 4. Каким образом ведут себя главный поток приложения и порожденные им потоки, после выполнения всех операций? 5. В чем заключаются отличия между классами TreeSet и HashSet? 6. На основе какой структуры данных реализован класс HashMap?

**Оценочные средства для проведения промежуточной аттестации
по дисциплине «Основы объектно-ориентированного программирования»**

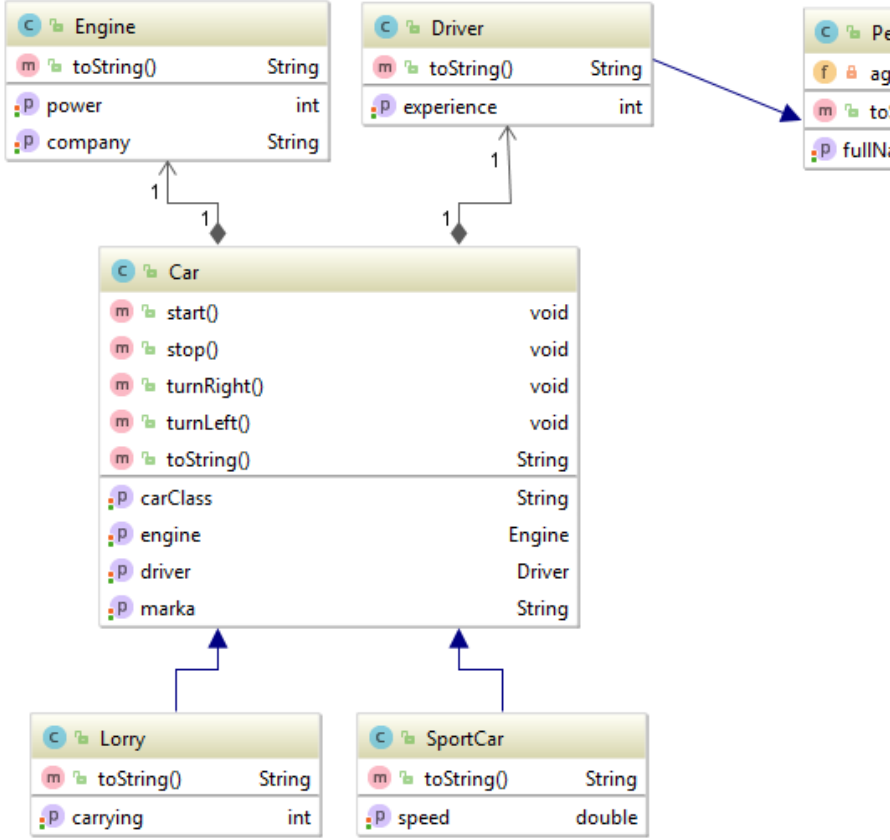
а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
ОПК-6: Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности		
ОПК-6.1	Разрабатывает алгоритмы и программы для практического применения в сфере своей профессиональной деятельности	Перечень теоретических вопросов к зачету: 1. Язык JAVA как объектно-ориентированный язык программирования. 2. Сущность и основные принципы языка JAVA. 3. Понятие платформы программирования. 4. Платформа J2SE. Основные достоинства, недостатки и особенности языка JAVA. 5. Основные области применения языка Java. 6. Основные понятия языка Java. Классы, объекты, интерфейсы, пакеты. 7. Структура файла исходного кода. 8. Виды приложений на языке Java. 9. Структура приложения в платформе J2SE. 10. Пример приложения на языке Java. Компиляция и отладка приложений. 11. Основные элементы языка. Лексемы. Идентификаторы. Ключевые слова. 12. Основные элементы языка. Константы. 13. Основные элементы языка. Примитивные типы данных – целые, символьный, вещественные, логические. 14. Основные элементы языка. Объявление и инициализация переменных. Время жизни переменных. 15. Операторы языка Java. Приоритет и правила ассоциативности. Порядок вычисления операндов. 16. Оператор присваивания =. Присваивание примитивных значений, присваивание ссылок, многократные присваивания. 17. Арифметические операторы. Приоритет и ассоциативность арифметических операторов. 18. Арифметические операторы. Порядок вычислений в арифметических выражениях. 19. Арифметические операторы. Диапазон числовых значений. 20. Арифметические операторы. Унарные и бинарные арифметические операторы. 21. Арифметические операторы. Составные операторы

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		присваивания. Операторы декремента и инкремента. 22. Операторы отношения. Равенство. Равенство значений примитивных типов данных. 23. Равенство ссылок на объекты. Равенство значений объектов. 24. Булевы логические операторы. Логические составные операторы присваивания. Условные операторы && и . Условный оператор ?/ 25. Целочисленные поразрядные операторы. Операторы ~,&, ,^. Операторы сдвига <<, >>, >>>. 26. Унарный оператор приведения (тип). 27. Преобразования сужения и расширения. 28. Числовые расширения. 29. Примеры преобразования типов. Неявное преобразование типов. Правила приведения. 30. Массивы. Объявление переменных -массивов. 31. Создание массива. Инициализация массива. 32. Использование массивов. Анонимные массивы. Многомерные массивы. 33. Операторы ветвления – сокращенный оператор if. 34. Оператор if – else. 35. Оператор switch. 36. Операторы цикла. Оператор while. Оператор do-while. 37. Оператор цикла for. 38. Команды перехода. Маркированные операторы. Операторы break, continue, return. 39. Определение классов. Модификаторы объявления классов. 40. Структура класса – поля, методы, конструкторы, блоки инициализации. 41. Определение полей. Модификаторы объявления полей. Инициализация значений. 42. Блоки инициализации. 43. Неизменяемые поля. 44. Методы класса. Модификаторы объявления методов классов. 45. Методы класса. Передача параметров. 46. Статические методы. 47. Неизменяемые методы. 48. Абстрактные методы. 49. Модификаторы и правила видимости. 50. Объектная ссылка this. 51. Перегрузка методов. 52. Конструкторы. 53. Конструктор по умолчанию. Перегруженные конструкторы. 54. Блоки инициализации. 55. Объекты. Создание объектов. Время жизни объекта и сборка мусора. 56. Одиночное наследование. Переопределение и скрытие методов. 57. Соккрытие (инкапсуляция) полей; 58. Соккрытие статического метода. 59. Объектная ссылка super. Организация цепочки конструкторов. 60. Интерфейсы.

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		61. Вложенные классы и интерфейсы. 62. Назначение исключительных ситуаций. 63. Типы исключительных ситуаций. 64. Исключительная ситуация как объект. 65. Основные классы исключительных ситуаций. Классы Exception, RuntimeException, Error. 66. Определение новых классов исключительных ситуаций. 67. Обработка исключений – try, catch, finally. Оператор throw. Генерация исключительной ситуации. 68. Диагностические утверждения. 69. Основные классы пакета java.lang. Класс Object. 70. Классы оболочки над примитивными типами. 71. неизменяемые строки - класс String. 72. Изменяемые строки – класс StringBuffer. 73. Класс Math – реализация основных математических функций. 74. Инструментальный набор коллекций. Классы и интерфейсы для реализации коллекций и карт. 75. Коллекции – основные действия, массовые операции, операции с массивами, итераторы. 76. Множества – классы HashSet и LinkedHashSet. 77. Списки- ArrayList, LinkedList 78. Карты. Классы HashMap, LinkedHashMap и Hashtable. 79. Отсортированные множества и отсортированные карты. 80. Интерфейсы Comparator, Comparable, SortedSet, SortedMap. 81. Работа с коллекциями. 82. Организация чтения/записи байтовых потоков. 83. Классы InputStreamReader и OutputStreamReader. 84. Буферизованные потоки чтения и записи – класс BufferedReader. 85. Чтение и запись из файлов. 86. Организация взаимодействия с консолью. 87. Многозадачность. Обзор потоков. 88. Главный поток. 89. Создание потока, реализация интерфейса Runnable.____ Примеры практических заданий для зачета: 1. Определить класс Reader, хранящий такую информацию о пользователе библиотеки: ФИО, номер читательского билета, факультет, дата рождения, телефон. Методы takeBook(), returnBook(). Разработать программу, в которой создается массив объектов данного класса. Перегрузить методы takeBook(), returnBook(): - takeBook, который будет принимать количество взятых книг. Выводит на консоль сообщение «Петров В. В. взял 3 книги». - takeBook, который будет принимать переменное количество названий книг. Выводит на консоль сообщение «Петров В. В. взял книги: Приключения, Словарь, Энциклопедия». - takeBook, который будет принимать переменное количество объектов класса Book (создать новый класс, содержащий имя

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		и автора книги). Выводит на консоль сообщение «Петров В. В. взял книги: Приключения, Словарь, Энциклопедия». Аналогичным образом перегрузить метод <code>returnBook()</code>. Выводит на консоль сообщение «Петров В. В. вернул книги: Приключения, Словарь, Энциклопедия». Или «Петров В. В. вернул 3 книги». 2. Создайте пример наследования, реализуйте класс <code>Student</code> и класс <code>Aspirant</code>, аспирант отличается от студента наличием некой научной работы. а) Класс <code>Student</code> содержит переменные: <code>String firstName</code>, <code>lastName</code>, <code>group</code>. А также <code>double averageMark</code>, содержащую среднюю оценку. б) Создать переменную типа <code>Student</code>, которая ссылается на объект типа <code>Aspirant</code>. в) Создать метод <code>getScholarship()</code> для класса <code>Student</code>, который возвращает сумму стипендии. Если средняя оценка студента равна 5, то сумма 100 грн, иначе 80. Переопределить этот метод в классе <code>Aspirant</code>. Если средняя оценка аспиранта равна 5, то сумма 200 грн, иначе 180. г) Создать массив типа <code>Student</code>, содержащий объекты класса <code>Student</code> и <code>Aspirant</code>. Вызвать метод <code>getScholarship()</code> для каждого элемента массива. 3. Создать класс <code>Car</code> в пакете <code>com.company.vehicles</code>, <code>Engine</code> в пакете <code>com.company.details</code> и <code>Driver</code> в пакете <code>com.company.professions</code>. Класс <code>Driver</code> содержит поля - ФИО, стаж вождения. Класс <code>Engine</code> содержит поля - мощность, производитель. Класс <code>Car</code> содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа <code>Driver</code>, мотор типа <code>Engine</code>. Методы <code>start()</code>, <code>stop()</code>, <code>turnRight()</code>, <code>turnLeft()</code>, которые выводят на печать: «Поехали», «Останавливаемся», «Поворот направо» или «Поворот налево». А также метод <code>printInfo()</code>, который выводит полную информацию об автомобиле, ее водителе и моторе. Создать производный от <code>Car</code> класс - <code>Loggy</code> (грузовик), характеризующийся также грузоподъемностью кузова. Создать производный от <code>Car</code> класс - <code>SportCar</code>, характеризующийся также предельной скоростью. Пусть класс <code>Driver</code> расширяет класс <code>Person</code>.

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		 <pre> classDiagram class Engine { +String toString() +int power +String company } class Driver { +String toString() +int experience } class Car { +void start() +void stop() +void turnRight() +void turnLeft() +String toString() +String carClass +Engine engine +Driver driver +String marka } class Lorry { +String toString() +int carrying } class SportCar { +String toString() +double speed } Engine "1" -- "1" Car Driver "1" -- "1" Car Car < -- Lorry Car < -- SportCar </pre> 4. Создать класс Animal и расширяющие его классы Dog, Cat, Horse. Класс Animal содержит переменные food, location и методы makeNoise, eat, sleep. Метод makeNoise, например, может выводить на консоль «<i>Такое-то животное спит</i>». Dog, Cat, Horse переопределяют методы makeNoise, eat. Добавьте переменные в классы Dog, Cat, Horse, характеризующие только этих животных. Создайте класс Ветеринар, в котором определите метод void treatAnimal(Animal animal). Пусть этот метод распечатывает food и location пришедшего на прием животного. В методе main создайте массив типа Animal, в который запишите животных всех имеющихся у вас типов. В цикле отправляйте их на прием к ветеринару. 5. Создайте суперкласс Shape и его наследники Circle, Rectangle. Класс Shape содержит абстрактный метод draw() и переменную хранящую цвет. Классы Circle, Rectangle содержат координаты точек. Создать массив содержащий эти фигуры. В цикле нарисовать их (вызвать метод draw). Добавить метод equals() для классов Shape, Circle, Rectangle. а) Создать класс Товар, имеющий переменные имя, цена, рейтинг. б) Создать класс Категория, имеющий переменные имя и

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		массив товаров. Создать несколько объектов класса Категория. в) Создать класс Basket, содержащий массив купленных товаров. г) Создать класс User, содержащий логин, пароль и объект класса Basket. Создать объект класса User. (Интернет магазин) Перечень практических работ: 1. Основы языка Java. Массивы, примитивные типы, объявление классов 2. Основы Языка Java. Перегрузка и перекрытие методов, 3. Наследование 4. Основы Языка Java. Перегрузка и перекрытие методов, наследование. Классы-оболочки 5. Основы Языка Java. Наследование, тригонометрические функции класса Math 6. Основы языка Java. Работа с изменяемыми и неизменяемыми строками 7. Основы языка Java. Наследование. Сравнение объектов 8. Основы языка Java. Наследование. Сравнение объектов. 9. Запись в файловый поток 10. ООП в JAVA, наследование, сериализация, файловые потоки ввода-вывода 11. ООП в Java. Наследование 12. Обработка изменяемых строк, коллекции, карты 13. Коллекции и списки 14. Наследование. Стандартные потоки ввода-вывода 15. Файлы. Файловые потоки ввода-вывода. Операции с файлами 16. Многопоточные приложения. Синхронизация

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «**Основы объектно-ориентированного программирования**» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета.

Зачет проводится в устной форме по теоретическим вопросам и задачам.

Критерии оценки (в соответствии с формируемыми компетенциями и планируемыми результатами обучения):

– на оценку «зачтено» – студент должен раскрыть содержание материала в объеме программы дисциплины, правильно дать необходимые определения, привести доказательства на основе математических и логических выкладок. Ответ должен быть самостоятельным, при ответе использованы знания, приобретённые ранее.

– на оценку «незачтено» – студент не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач, основное содержание учебного материала не раскрыто. При ответе допущены грубые ошибки в определениях, доказательства теорем не проведено, не даны ответы на дополнительные вопросы преподавателя, отсутствуют навыки исследовательской деятельности.