



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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



УТВЕРЖДАЮ
Директор ИЭиАС
С.И. Лукьянов

26.02.2020 г.

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
Программа подготовки - академический бакалавриат

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Автоматизированных систем управления
Курс	2
Семестр	3

Магнитогорск
2020 год

Рабочая программа составлена на основе ФГОС ВО по направлению подготовки 27.03.04 УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ (уровень бакалавриата) (приказ Минобрнауки России от 20.10.2015 г. № 1171)

Рабочая программа рассмотрена и одобрена на заседании кафедры Автоматизированных систем управления
12.02.2020, протокол № 6

Зав. кафедрой  С.М. Андреев

Рабочая программа одобрена методической комиссией ИЭиАС
26.02.2020 г. протокол № 5

Председатель  С.И. Лукьянов

Рабочая программа составлена:

доцент кафедры АСУ, канд. техн. наук  Е.С. Рябчикова

Рецензент:

зам. директора ЗАО "КонсОМ СКС" , канд. техн. наук
 Ю.Н. Волщук



Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 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 2022 - 2023 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 2023 - 2024 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 вариативную часть учебного плана образовательной программы.

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

Информатика и информационные технологии

Программирование и основы алгоритмизации

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

Операционные системы реального времени

Интегрированные системы проектирования и 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:

Структурный элемент компетенции	Планируемые результаты обучения
ДПК-3 способностью разрабатывать новое программное обеспечение, необходимое для управления техническими системами и решения практических задач профессиональной деятельности	
Знать	<ul style="list-style-type: none">- базовые понятия и синтаксис языка, технологию ООП и приемы разработки программ;- методы определения и использования основных объектов и конструкций языка;- технологию организации и использования иерархии классов, предопределенных классов и типов данных, методы ограничения доступа и обработки исключительных ситуаций;- методы параметризации классов и их использование для решения задач;- методы применения шаблонов и контейнерных абстракций;- работу с потоками и разработку многопоточных приложений;

Уметь	<ul style="list-style-type: none">- определять абстракции, модули, строить иерархию классов для реализации программ;- использовать методы: типизации, инкапсуляции, наследования, полиморфизма для разработки программных продуктов;- использовать возможности стандартных библиотек;- использовать механизм исключений для создания устойчивых приложений;- создавать свои и использовать предоставляемые стандартные библиотеки шаблонов сложных структур данных;- использовать технологию ООП для разработки сложных программ и систем;
Владеть	<ul style="list-style-type: none">- методами и инструментальными средствами и системами разработки объектно-ориентированных программ;- техникой создания объектно-ориентированных программных компонент и организацией их взаимодействия в программных проектах.

4. Структура, объём и содержание дисциплины (модуля)

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

- контактная работа – 37 акад. часов;
- аудиторная – 36 акад. часов;
- внеаудиторная – 1 акад. часов
- самостоятельная работа – 35 акад. часов;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	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	2			2	Самостоятельное изучение учебной литературы	Собеседование	ДПК-3
1.2 Конструкторы. Модификаторы управления доступа и области видимости. Скрытие информации и инкапсуляция.		2		2	4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ДПК-3
1.3 Геттеры и сеттеры. Ключевое слово «this». Метод «toString». Константы		1		2	4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ДПК-3
Итого по разделу		5		4	10			
2. Наследование								
2.1 Области видимости. Переопределение методов и сокрытие полей.	3	2			2	Самостоятельное изучение учебной литературы.	Собеседование	ДПК-3
2.2 Ключевое слово «super». Конструктор без параметров по умолчанию.		2		2	4	Самостоятельное изучение учебной литературы. Выполнение	Собеседование	ДПК-3
2.3 Одиночное наследование		1		2	4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ДПК-3

Итого по разделу		5		4	10			
3. Полиморфизм, абстрактные классы и интерфейсы								
3.1 Подстановка	3	1			2	Самостоятельное изучение учебной литературы.	Собеседование	ДПК-3
3.2 Апкастинг и даункастинг.		1		2	4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ДПК-3
3.3 Оператор «instanceof»		2		2	4	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	ДПК-3
Итого по разделу		4		4	10			
4. Абстрактные классы и интерфейсы								
4.1 Абстрактный метод. Абстрактный класс. Интерфейс	3			2	1	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	
4.2 Реализация множественных интерфейсов. Интерфейс и абстрактный суперкласс				2	2	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	
4.3 Динамическое связывание. Инкапсуляция, связывание и связность.		4		2	2	Самостоятельное изучение учебной литературы. Выполнение практической работы	Собеседование Устный опрос по практической работе	
Итого по разделу		4		6	5			
Итого за семестр		18		18	35		зачёт	
Итого по дисциплине		18		18	35		зачет	ДПК-3

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 (модуля)

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: 17.09.2020). – Режим доступа: по подписке.

б) Дополнительная литература:

1. Ашарина И. В. Объектно-ориентированное программирование в С++: лекции и упражнения. Учебное пособие для вузов / И.В. Ашарина. - Москва : Горячая Линия–Телеком, 2012. - 320 с. - ISBN 978-5-9912-7001-4. - URL: <https://ibooks.ru/reading.php?productid=333353> (дата обращения: 17.09.2020). - Текст: электронный.

2. Хабибуллин И. Программирование на языке высокого уровня. С/С++ / И. Хабибуллин. - Санкт-Петербург : БХВ-Петербург, 2010. - 512 с. - ISBN 5-94157-559-9. - URL: <https://ibooks.ru/reading.php?productid=18532> (дата обращения: 17.09.2020). - Текст: электронный.

3. Кубенский А. Структуры и алгоритмы обработки данных: объектно ориентированный подход и реализация на С++ / А. Кубенский. - Санкт-Петербург : БХВ-Петербург, 2010. - 464 с. - ISBN 5-94157-506-8. - URL: <https://ibooks.ru/reading.php?productid=18563> (дата обращения: 17.09.2020). - Текст: электронный.

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: 17.09.2020). – Режим доступа: по подписке.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Windows XP Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Office 2003 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое ПО	бессрочно
FAR Manager	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	URL: https://elibrary.ru/project_risc.asp
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/

Информационная система - Единое окно доступа к информационным ресурсам	URL: http://window.edu.ru/
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	http://magtu.ru:8085/marcweb2/Default.asp
Федеральный образовательный портал – Экономика. Социология. Менеджмент	http://ecsocman.hse.ru/
Университетская информационная система РОССИЯ	https://uisrussia.msu.ru
Международная наукометрическая реферативная и полнотекстовая база данных научных изданий «Web of science»	http://webofscience.com
Международная реферативная и полнотекстовая справочная база данных	http://scopus.com
Международная база полнотекстовых журналов Springer Journals	http://link.springer.com/
Международная коллекция научных протоколов по различным отраслям знаний	http://www.springerprotocols.com/

Международная база научных материалов в области физических наук и инжиниринга	http://materials.springer.com/
Международная база справочных изданий по всем отраслям знаний SpringerReference	http://www.springer.com/references
Международная реферативная база данных по чистой и прикладной математике zbMATH	http://zbmath.org/
Международная реферативная и полнотекстовая справочная база данных научных изданий «Springer Nature»	https://www.nature.com/siteindex
Архив научных журналов «Национальный электронно-информационный конкорциум» (НП НЭИКОН)	https://archive.neicon.ru/xmlui/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Учебные аудитории для проведения занятий лекционного типа
Мультимедийные средства хранения, передачи и представления информации
2. Учебная аудитория для проведения практических занятий: компьютерный класс
Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
3. Помещения для самостоятельной работы обучающихся
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
Доска, мультимедийный проектор, экран
5. Помещение для хранения и профилактического обслуживания учебного оборудования
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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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»**

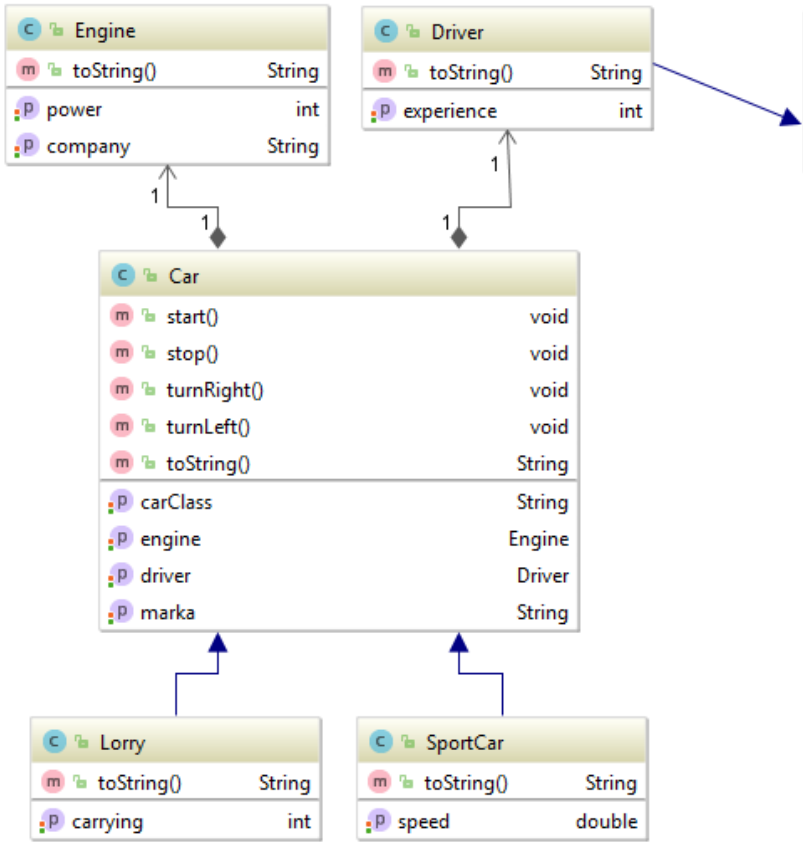
а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
ДПК-3: способностью разрабатывать новое программное обеспечение, необходимое для управления техническими системами и решения практических задач профессиональной деятельности		
Знать	<ul style="list-style-type: none"> – базовые понятия и синтаксис языка, технологию ООП и приемы разработки программ; – методы определения и использования основных объектов и конструкций языка; – технологию организации и использования иерархии классов, предопределенных классов и типов данных, методы ограничения доступа и обработки исключительных ситуаций; – методы параметризации классов и их использование для решения задач; – методы применения шаблонов и контейнерных абстракций; – работу с потоками и разработку многопоточных приложений; 	<p><i>Перечень теоретических вопросов к зачету:</i></p> <ol style="list-style-type: none"> 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
		<p>присваивания. Операторы декремента и инкремента.</p> <p>22. Операторы отношения. Равенство. Равенство значений примитивных типов данных.</p> <p>23. Равенство ссылок на объекты. Равенство значений объектов.</p> <p>24. Булевы логические операторы. Логические составные операторы присваивания. Условные операторы && и . Условный оператор ?/</p> <p>25. Целочисленные поразрядные операторы. Операторы ~, &, , ^ . Операторы сдвига <<, >>, >>>.</p> <p>26. Унарный оператор приведения (тип) .</p> <p>27. Преобразования сужения и расширения.</p> <p>28. Числовые расширения.</p> <p>29. Примеры преобразования типов. Неявное преобразование типов.</p> <p>Правила приведения.</p> <p>30. Массивы. Объявление переменных -массивов.</p> <p>31. Создание массива. Инициализация массива.</p> <p>32. Использование массивов. Анонимные массивы. Многомерные массивы.</p> <p>33. Операторы ветвления – сокращенный оператор if.</p> <p>34. Оператор if – else.</p> <p>35. Оператор switch.</p> <p>36. Операторы цикла. Оператор while. Оператор do-while.</p> <p>37. Оператор цикла for.</p> <p>38. Команды перехода. Маркированные операторы. Операторы break, continue, return.</p> <p>39. Определение классов. Модификаторы объявления классов.</p> <p>40. Структура класса – поля, методы, конструкторы, блоки инициализации.</p> <p>41. Определение полей. Модификаторы объявления полей. Инициализация значений.</p> <p>42. Блоки инициализации.</p> <p>43. Неизменяемые поля.</p> <p>44. Методы класса. Модификаторы объявления методов классов.</p> <p>45. Методы класса. Передача параметров.</p> <p>46. Статические методы.</p> <p>47. Неизменяемые методы.</p> <p>48. Абстрактные методы.</p> <p>49. Модификаторы и правила видимости.</p> <p>50. Объектная ссылка this.</p> <p>51. Перегрузка методов.</p> <p>52. Конструкторы.</p> <p>53. Конструктор по умолчанию. Перегруженные конструкторы.</p> <p>54. Блоки инициализации.</p> <p>55. Объекты. Создание объектов. Время жизни объекта и сборка мусора.</p> <p>56. Одиночное наследование. Переопределение и скрытие</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>методов.</p> <p>57. Соккрытие (инкапсуляция) полей;</p> <p>58. Соккрытие статического метода.</p> <p>59. Объектная ссылка super. Организация цепочки конструкторов.</p> <p>60. Интерфейсы.</p> <p>61. Вложенные классы и интерфейсы.</p> <p>62. Назначение исключительных ситуаций.</p> <p>63. Типы исключительных ситуаций.</p> <p>64. Исключительная ситуация как объект.</p> <p>65. Основные классы исключительных ситуаций. Классы Exception, RuntimeException, Error.</p> <p>66. Определение новых классов исключительных ситуаций.</p> <p>67. Обработка исключений – try, catch, finally. Оператор throw. Генерация исключительной ситуации.</p> <p>68. Диагностические утверждения.</p> <p>69. Основные классы пакета java.lang. Класс Object.</p> <p>70. Классы оболочки над примитивными типами.</p> <p>71. Неизменяемые строки - класс String.</p> <p>72. Изменяемые строки – класс StringBuffer.</p> <p>73. Класс Math – реализация основных математических функций.</p> <p>74. Инструментальный набор коллекций. Классы и интерфейсы для реализации коллекций и карт.</p> <p>75. Коллекции – основные действия, массовые операции, операции с массивами, итераторы.</p> <p>76. Множества – классы HashSet и LinkedHashSet.</p> <p>77. Списки- ArrayList, LinkedList</p> <p>78. Карты. Классы HashMap, LinkedHashMap и Hashtable.</p> <p>79. Отсортированные множества и отсортированные карты.</p> <p>80. Интерфейсы Comparator, Comparable, SortedSet, SortedMap.</p> <p>81. Работа с коллекциями.</p> <p>82. Организация чтения/записи байтовых потоков.</p> <p>83. Классы InputStreamReader и OutputStreamReader.</p> <p>84. Буферизованные потоки чтения и записи – класс BufferedReader.</p> <p>85. Чтение и запись из файлов.</p> <p>86. Организация взаимодействия с консолью.</p> <p>87. Многозадачность. Обзор потоков.</p> <p>88. Главный поток.</p> <p>89. Создание потока, реализация интерфейса Runnable.</p>
Уметь	–определять абстракции, модули, строить иерархию классов	<p>Примеры практических заданий для зачета:</p> <p>1. Определить класс Reader, хранящий такую информацию о пользователе библиотеки: ФИО, номер читательского</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>для реализации программ;</p> <p>–использовать методы: типизации, инкапсуляции, наследования, полиморфизма для разработки программных продуктов;</p> <p>–использовать возможности стандартных библиотек;</p> <p>–использовать механизм исключений для создания устойчивых приложений;</p> <p>–создавать свои и использовать предоставляемые стандартные библиотеки шаблонов сложных структур данных;</p> <p>–использовать технологию ООП для разработки сложных программ и систем;</p>	<p>билета, факультет, дата рождения, телефон. Методы takeBook(), returnBook(). Разработать программу, в которой создается массив объектов данного класса. Перегрузить методы takeBook(), returnBook():</p> <ul style="list-style-type: none"> - 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 и автора книги). Выводит на консоль сообщение «Петров В. В. взял книги: Приключения, Словарь, Энциклопедия». <p>Аналогичным образом перегрузить метод returnBook(). Выводит на консоль сообщение «Петров В. В. вернул книги: Приключения, Словарь, Энциклопедия». Или «Петров В. В. вернул 3 книги».</p> <p>2. Создайте пример наследования, реализуйте класс Student и класс Aspirant, аспирант отличается от студента наличием некой научной работы.</p> <p>а) Класс Student содержит переменные: String firstName, lastName, group. А также double averageMark, содержащую среднюю оценку.</p> <p>б) Создать переменную типа Student, которая ссылается на объект типа Aspirant.</p> <p>в) Создать метод getScholarship() для класса Student, который возвращает сумму стипендии. Если средняя оценка студента равна 5, то сумма 100 грн, иначе 80. Переопределить этот метод в классе Aspirant. Если средняя оценка аспиранта равна 5, то сумма 200 грн, иначе 180.</p> <p>г) Создать массив типа Student, содержащий объекты класса Student и Aspirant. Вызвать метод getScholarship() для каждого элемента массива.</p> <p>3. Создать класс Car в пакете com.company.vehicles, Engine в пакете com.company.details и Driver в пакете com.company.professions. Класс Driver содержит поля - ФИО, стаж вождения. Класс Engine содержит поля - мощность, производитель. Класс Car содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа Driver, мотор типа Engine. Методы start(), stop(), turnRight(), turnLeft(), которые выводят на печать: «Поехали», «Останавливаемся», «Поворот направо» или «Поворот налево». А также метод printInfo(), который выводит полную информацию об</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>автомобиле, ее водителе и моторе. Создать производный от Car класс - Lorry (грузовик), характеризуемый также грузоподъемностью кузова. Создать производный от Car класс - SportCar, характеризуемый также предельной скоростью. Пусть класс Driver расширяет класс Person.</p>  <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> <p>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.</p> <p>5. Создайте суперкласс Shape и его</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>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.</p> <p>б. а) Создать класс Товар, имеющий переменные имя, цена, рейтинг. б) Создать класс Категория, имеющий переменные имя и 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)</p>
Владеть	<p>–методами и инструментальным и средствами и системами разработки объектно-ориентированных программ; –техникой создания объектно-ориентированных программных компонент и организацией их взаимодействия в программных проектах.</p>	<p>Перечень практических работ:</p> <ol style="list-style-type: none"> 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.