

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»
ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра Компьютерных технологий

УТВЕРЖДАЮ:

проректор по научно-методической
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ»

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Профиль подготовки: **Информатика и вычислительная техника**

Образовательная программа: **бакалавриат**

Квалификация: **академический бакалавр**

Форма обучения: **очная, очно-заочная, заочная, в том числе с ускоренным сроком обучения**

Донецк 2020

УТВЕРЖДАЮ:

Декан физико-технического факультета

Фоменко С.А.

«17» апреля 2020 г.

М.П.



Программа учебной дисциплины **«Объектно-ориентированное программирование»** составлена на основе Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённого приказом Министерства образования и науки ДНР от «21» января 2016 г. №31»; «Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики», утверждённого приказом Министерства образования и науки ДНР №1171 от «10» ноября 2017 г.»; учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 09.03.01 Информатика и вычислительная техника, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Старший преподаватель кафедры компьютерных технологий

Гукай А.Е.

Программа учебной дисциплины утверждена на заседании кафедры компьютерных технологий

Протокол № 12 от «2» апреля 2020 г.

Зав. кафедрой компьютерных технологий

Ермоленко Т.В.

Программа учебной дисциплины одобрена учебно-методической комиссией физико-технического факультета

Протокол № 5 от «15» апреля 2020 г.

Председатель учебно-методической комиссии физико-технического факультета

Котенко В.Н

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Объектно-ориентированное программирование» относится к вариативной части профессионального блока и состоит из четырёх содержательных модулей:

модуль 1 – «Теоретические основы ООП», модуль 2 – «Методы и механизм построения приложений с графическим интерфейсом», модуль 3 – «Основы программирования на языке Android.», модуль 4 – «Углубленное программирование на языке Android.».

Основывается на базе дисциплин: «Основы программирования», «Информатика и информационно-коммуникационные технологии», «Архитектура ЭВМ и микроконтроллеров», «ЭВМ и периферийные устройства». Изучение дисциплины «Объектно-ориентированное программирование» является основой для изучения дальнейших дисциплин, использующих ЭВМ и программирование, таких как «Разработка программных комплексов», «Программная инженерия».

2. СТРУКТУРА ДИСЦИПЛИНЫ

Характеристика учебной дисциплины	очная форма обучения на базе		заочная форма обучения на базе		
	СОО	СПО (сокращ.)	СОО	СПО (сокращ.)	ВПО (сокращ.)
Образовательный уровень	Бакалавр				
Направление подготовки	09.03.01 Информатика и вычислительная техника				
Профиль	Информатика и вычислительная техника				
Количество содержательных модулей (тем)	4 (24)				
Дисциплина базовой / вариативной части образовательной программы ¹	Профессиональный блок. Вариативная часть				
Формы контроля	текущие, 2 мод. контроля, диф. зачёт, экзамен				
Показатели	очная форма обучения на базе		заочная форма обучения на базе		
	СОО	СПО (сокращ.)	СОО	СПО (сокращ.)	ВПО (сокращ.)
Количество зачётных единиц (кредитов)	8	8	8	8	8
Количество часов	288	288	288	288	288
Год подготовки	4	3	4	3	4
Семестр	7,8	5,6	7,8	7,8	7,8
Количество часов					
- лекционных	46	46	10	10	10
- практических, семинарских					
- лабораторных	56	56	14	14	14
- самостоятельной работы	186	186	264	264	264
в т.ч. индивидуальное задание					
Недельное количество часов, т.ч.	6	6	6	6	6
аудиторных	4 и 3	4 и 3	1 и 0,77	1 и 0,77	1 и 0,77

СОО – среднее общее образование

СПО – среднее профессиональное образование

ВПО – высшее профессиональное образование

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Подготовка современного специалиста требует уверенного владения возможностями, предоставляемыми компьютерными технологиями. Изучение настоящей дисциплины обеспечивает подготовку специалиста, владеющего фундаментальными знаниями и практическими навыками в области основ объектно-ориентированного анализа, программирования и элементов проектирования при решении практических задач.

Цели и задачи.

Цель – формирование системного представления о принципах построения абстрактных типов данных, применения их для решения практических задач, углубление знаний в области алгоритмизации и приемов программирования на языках высокого уровня, получение практических навыков проектирования и реализации сложных программных продуктов.

Задачи – изучения дисциплины является формирование у студентов теоретических знаний и практических навыков объектно-ориентированного программирования с использованием языка высокого уровня Java, изучение основных принципов объектно-ориентированного программирования, изучение стандартных библиотек классов и шаблонов; привитие устойчивых практических навыков применения современных инструментальных средств при разработке программного обеспечения.

Требования к результатам освоения дисциплины:

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО по данному направлению подготовки (профилю):

а) общекультурных (ОК):

способность к самоорганизации и самообразованию (ОК-7).

б) общепрофессиональных (ОПК):

основательная подготовка по математике для использования математического аппарата при решении прикладных и научных задач в области компьютерной инженерии (ОПК-1);

знание современных методов построения и анализа алгоритмов, основ численных методов и умение их использовать на практике (ОПК-4).

в) профессиональных (ПК):

проектно-конструкторская деятельность:

знать архитектуру компьютеров, уметь применять их в процессе эксплуатации (ПК-1);

пользоваться методиками использования программных средств для решения практических задач (ПК-2);

использовать и самостоятельно разрабатывать интерфейсы взаимодействия человека и ЭВМ (ПК-3);

знание принципов программирования, средств современных языков программирования, структур данных (ПК-5);

проектно-технологическая деятельность:

знание методологических принципов построения современных компьютерных систем разной организации для высокопродуктивной обработки информации (ПК-12);

знание теоретических (логических и арифметических) основ построения современных компьютеров и умение их использовать при решении профессиональных задач (ПК-13);

знание современных технологий и инструментальных способов разработки сложных программных систем (инженерии программного обеспечения), умение их использовать на всех этапах жизненного цикла программ (ПК-14);

научно-исследовательская деятельность:

базовые знания научно-методических основ и стандартов в области компьютерной инженерии, проводить эксперимент по проверке корректности решений, рассчитывать экономическую эффективность (ПК-15);

умение готовить и проводить доклады с использованием современных компьютерных средств, писать научно-технические отчёты, оформлять результаты исследований в виде статей (ПК-16);

педагогическая деятельность:

готовить конспекты лекций, проводить повышение квалификации сотрудников (ПК-17);

сервисно-эксплуатационная деятельность:

инсталлировать, настраивать и сопровождать программное и аппаратное обеспечение информационных и автоматизированных систем (ПК-21).

В результате изучения учебной дисциплины студент должен

Знать:

- базовые понятия и синтаксис языка, технологию объектно-ориентированного программирования;
- методы определения и использования основных объектов и конструкций языка;
- основные технологические приемы разработки программ;
- технологию организации и использования иерархии классов, предопределенных классов и типов данных;
- оптимальные методы использования объектно-ориентированного языка для решения прикладных задач по направлениям;
- принципы построения приложений с графическим интерфейсом;
- способы обмена информацией между приложениями по сети;
- Элементы управления, горячие клавиши, инструменты интегрированной среды разработки Android Studio;
- основы создания приложений для ОС Android;
- компоненты Android приложения и их активация;
- структуру файла манифеста;
- назначение ресурсов Android приложения;
- жизненный цикл Android приложения;
- возможности платформы для обеспечения совместимости устройств;
- объекты Intent и фильтры объектов Intent;

Уметь:

- использовать технологию объектно-ориентированного программирования;
- определять программные абстракции, модули, строить иерархию классов для реализации программ;
- создавать свои и использовать предоставляемые стандартные библиотеки шаблонов сложных структур данных;
- применять методы инкапсуляции, наследования, полиморфизма и другие сложные абстракции объектно-ориентированного программирования для разработки сложных программ и систем по направлениям;
- Использовать компоненты Android приложения;
- объявлять требования Android приложения;

- управлять жизненным циклом Android приложения;
- объявлять требования Android приложения;
- создавать динамический интерфейс пользователя с использованием фрагментов;
- использовать многопоточность для повышения производительности;
- определять положение и использовать датчики мобильного устройства;
- сохранять и передавать данные на сервер;
- воспроизводить медиа данные и использовать камеру мобильного устройства;
- использовать библиотеки при создании мобильных приложений;

Владеть:

- методами объектно-ориентированного программирования;
- пониманием особенностей языков программирования и нюансов выполнения, написанных на них программ, вытекающих из принятых традиционных способов организации трансляторов (компиляция и интерпретация, время связывания, статические и динамические свойства), и осознанно их использовать.

Программа рассчитана на объем 288 учебных часа, из них 105 – аудиторных. Примерное распределение аудиторных часов по видам занятий: лекций – 47 часов, лабораторных занятий – 58 часов. Самостоятельная работа студента – 183 часа.

4. Содержание дисциплины (модуля) и формы организации учебного процесса

Порядковый номер и тема	Краткое содержание темы
	<i>Содержательный модуль 1. Теоретические основы ООП</i>
Тема 1. Основы синтаксиса языка Java	Первая программа на языке Java. Переменные, базовые типы данных, условные операторы, циклы. Регулярные выражения.
Тема 2. Концептуальные основы объектно-ориентированного подхода. Классы и объекты.	Основные направления в программировании. Возникновение объектно-ориентированного программирования (ООП). Базовые принципы ООП. Основные положения объектной модели. Абстрагирование. Модульность. Иерархия. Типизация. классы и объекты. Инициализация и разрушение объекта. Компоненты класса. Конструкторы.
Тема 3. Методы и механизмы разработки объектно-ориентированных программ	Инкапсуляция и организация доступа к элементам класса: наследование; полиморфизм, его проявления и механизмы использования. Понятие раннего и позднего связывания. Абстрактные классы и интерфейсы, их назначение и свойства. Перегрузка методов классов
Тема 4. Исключения и их обработка	Основы обработки исключений. Генерация исключений. Перехватывание исключений. Повторная генерация исключения. Исключения и наследование. Спецификация исключений. Иерархия исключений.
Тема 5. Поток. Многопоточность	Потоки ввода\вывода. Организация работы с файлами. Понятие потоков. Жизненный цикл потока. Управление потоками, их приоритеты. Методы синхронизации потоков
Тема 6. Параметризация объектов в объектно-ориентированных программах	Введение в стандартную библиотеку шаблонов (классов коллекций), основные понятия. Классы контейнеры и итераторы. Типы контейнерных классов. Использование контейнерных классов. Шаблонные классы. Совместное использование параметризации и принципов наследования. Организация внешнего доступа к компонентам параметризованных классов. Списки. Стеки. Деки. Очереди.
	<i>Содержательный модуль 2. Методы и механизм построения приложений с графическим интерфейсом.</i>
Тема 7. Методы и механизмы построения оконного интерфейса.	SWING и AWT. Основы оконной графики. Элементы управления и их свойства. Способы размещения элементов управления. Менеджеры компоновки: border layout, box layout, card layout, flow layout. События и механизмы их обработки.
Тема 8. Работа с графикой	Графический контекст компонентов. Рисование графических примитивов. Работа со шрифтами и текстом при работе с графикой.
Тема 9.	Подключение баз данных к проектам на языке Java. Драйвера

Работа с базой данных в Java.	соединения с базой данных. Выполнение запросов к базе данных, получение результатов из базы данных. Объектно-реляционный подход к работе с базой данных. Hibernate.
Тема 10. Java веб	Сокеты и сервлеты. Maven. Настройка проекта веб-приложения. Dependency Injection. JSP и JSF. Spring MVC.
	Содержательный модуль 3. Основы программирования на языке Android.
Тема 11. Знакомство с операционной системой Android	История развития. Достоинства и недостатки ОС. Установка среды разработки Android Studio. Структура проекта. Отображаемые компоненты и их свойства.
Тема 12. Структура и жизненный цикл приложения	Ресурсы приложения. Обработка событий. Жизненный цикл Activity. Вызов системных приложений. Сохранение настроек приложения.
Тема 13. Работа с фрагментами	Назначение фрагментов. Передача данных между фрагментами. Взаимодействие с Activity.
Тема 14. Анимация и работа с сервисами	Планирование покадровой анимации, анимирование, использование класса Camera. Проверка безопасности, работа с сервисами.
	Содержательный модуль 4. Работа с данными в ОС Android
Тема 15. Отображение данных в списке	Назначение RecyclerView. Использование RecyclerViewAdapter. Data binding. Библиотека Glide.
Тема 16. Хранение данных	Работа с файлами. База данных Sqlite. Объектно-реляционное отображение. Библиотека Realm.
Тема 17. Rest Api. Доступ в интернет	Особенности работы с сетью. Асинхронные запросы. Реактивное программирование. Обработка xml и json при помощи библиотеки Retrofit.
Тема 18. Работа с Google Play	Подготовка AndroidManifest.xml для загрузки, локализация приложения, подготовка ярлыка приложения, подготовка APK файла для загрузки, работа пользователя с Google Play.

Курс дисциплины «Объектно-ориентированное программирование» предусматривает следующие **формы организации учебного процесса**:

1. лекции;
2. лабораторные занятия;
3. самостоятельная работа студента.

По источнику передачи и восприятия учебной информации используются словесные (лекция, беседа), наглядные (иллюстрация, демонстрация), практические (исследования, упражнения, лабораторные работы) методы.

По характеру познавательной деятельности студентов используются объяснительно-иллюстративные и репродуктивные методы, проблемное преподавание, частично-поисковый и исследовательский методы.

В зависимости от основной дидактической цели и задач используются методы устного изложения знаний, закрепление учебного материала, самостоятельной работы студентов по

осмыслению и усвоению нового материала, работы по применению знаний на практике и выработке умений и навыков, проверки и оценки знаний, умений и навыков.

Для промежуточного контроля по учебной дисциплине и диагностики компетенций студентов используются следующие формы:

- отчеты по аудиторным практическим упражнениям с их устной защитой;
- отчеты по домашним практическим упражнениям с их устной защитой;
- отчеты по лабораторным работам с их устной защитой;
- контрольные работы;
- проверка конспектов;
- устный опрос;
- электронные тесты;
- доклады на конференциях;
- оценивание на основе модульно-рейтинговой системы;
- итоговое тестирование.

Основные методы (технологии) обучения, отвечающие целям и задачам дисциплины:

- проблемное обучения (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемое на лекционных занятиях;
- учебно-исследовательская деятельность, творческий подход, реализуемые на лабораторных занятиях;
- проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении лабораторных работ.

Тематический план

	Содержательный модуль 1																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма						Заочная форма																
							на базе общего среднего образования						на базе среднего профессионального образования						на базе высшего профессионального образования				
	всего	в т.ч.					всего	в т.ч.					всего	в т.ч.					всего	в т.ч.			
лекции		практически	Лабораторные	самостоятельная работа	индивидуальная работа	лекции		практически	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практически	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практически	самостоятельная работа	индивидуальная работа	
Тема 1. Основы синтаксиса языка Java	10	2		2	6		13	0,5		0,5	12		13	0,5		0,5	12		13	0,5		0,5	12
Тема 2. Концептуальные основы объектно-ориентированного подхода. Классы и объекты.	10	2		2	6		13	0,5		0,5	12		13	0,5		0,5	12		13	0,5		0,5	12
Тема 3. Методы и механизмы разработки объектно-ориентированных программ	16	4		4	8		15,5	0,5		1	14		15,5	0,5		1	14		15,5	0,5		1	14
Тема 4. Исключения и их обработка	10	2		2	6		15	0,5		0,5	14		15	0,5		0,5	14		15	0,5		0,5	14
Тема 5. Потоки. Многопоточность.	16	4		4	8		15,5	0,5		1	14		15,5	0,5		1	14		15,5	0,5		1	14
Тема 6. Параметризация объектов в объектно-ориентированных программах	14	2		4	8		15	0,5		0,5	14		15	0,5		0,5	14		15	0,5		0,5	14
Итого по 1-му содержательному модулю	76	16		18	42		87	3		4	80		87	3		4	80		87	3		4	80

	Содержательный модуль 2																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма						Заочная форма																
							на базе общего среднего образования					на базе среднего профессионального образования					на базе высшего профессионального образования						
	всего	в т.ч.					всего	в т.ч.					всего	в т.ч.					всего	в т.ч.			
лекции		практические	Лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	самостоятельная работа	индивидуальная работа	
Тема 7. Методы и механизмы построения оконного интерфейса.	18	4		4	10		21,5	0,5		1	20		21,5	0,5		1	20		21,5	0,5		1	20
Тема 8. Работа с графикой	18	4		4	10		22	1		1	20		22	1		1	20		22	1		1	20
Тема 9. Работа с базой данных в Java.	14	2		2	10		21,5	0,5		1	20		21,5	0,5		1	20		21,5	0,5		1	20
Тема 10. Java веб	24	6		4	14		22	1		1	20		22	1		1	20		22	1		1	20
Итого по 2-му содержательному модулю	74	16		14	44		87	3		4	80		87	3		4	80		87	3		4	80

	Содержательный модуль 3																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма						Заочная форма																
							на базе общего среднего образования						на базе среднего профессионального образования						на базе высшего профессионального образования				
	всего	в т.ч.					всего	в т.ч.					всего	в т.ч.					всего	в т.ч.			
лекции		практические	Лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	самостоятельная работа	индивидуальная работа	
Тема 11. Знакомство с операционной системой Android	16	2		2	12		13,5	0,5		1	12		13,5	0,5		1	12		13,5	0,5		1	12
Тема 12. Структура и жизненный цикл приложения	18	2		4	12		13,5	0,5		1	12		13,5	0,5		1	12		13,5	0,5		1	12
Тема 13. Работа с фрагментами	18	2		4	12		14	1		1	12		14	1		1	12		14	1		1	12
Тема 14. Анимация и работа с сервисами	20	2		4	14		16	1		1	14		16	1		1	14		16	1		1	14
Итого по 3-му содержательному модулю	72	8		14	50		57	3		4	50		57	3		4	50		57	3		4	50

	Содержательный модуль 4																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма						Заочная форма																
							на базе общего среднего образования						на базе среднего профессионального образования						на базе высшего профессионального образования				
	всего	в т.ч.					всего	в т.ч.					всего	в т.ч.					всего	в т.ч.			
Лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	лабораторные	самостоятельная работа	индивидуальная работа	лекции		практические	самостоятельная работа	индивидуальная работа	
Тема 15. Отображение данных в списке	16	2		2	12		13,5	0,5		1	12		13,5	0,5		1	12		13,5	0,5		1	12
Тема 16. Хранение данных	16	2		2	12		13,5	0,5		1	12		13,5	0,5		1	12		13,5	0,5		1	12
Тема 17. Rest Api. Доступ в интернет	18	2		4	12		14	1		1	12		14	1		1	12		14	1		1	12
Тема 18. Работа с Google Play	16	1		4	11		16	1		1	14		16	1		1	14		16	1		1	14
Итого по 4-му содержательному модулю	66	7		12	47		57	3		4	50		57	3		4	50		57	3		4	50
Всего часов	288	47		58	183		288	12		16	260		288	12		16	260		288	12		16	260

6. Темы семинарских занятий

Семинарские занятия не предусмотрены

7. Темы практических занятий.

Практические занятия не предусмотрены

8. Темы лабораторных занятий.

Семестр 1

1. Консольные приложения.
2. Работа с файловой системой в Java. Регулярные выражения
3. Swing и AWT. Графический интерфейс пользователя
4. Работа с графикой в Java.
5. Использование шаблонов (параметризованных) классов.
6. Разработка игры с применением навыков ООП.
7. Сетевые java-приложения. Многопоточность.
8. Использование баз данных в приложениях Java.
9. Разработка веб-сервисов на Java. Сервлеты
10. Возможности Java EE для разработки веб-приложения.

Семестр 2

1. Введение. Структура Android-проекта.
2. Виды Layouts. Ключевые отличия и свойства.
3. Обработчики событий.
4. Создание и вызов Activity.
5. Хранение данных. Preferences.
6. Хранение данных. SQLite
7. События в ListView.
8. Обзор адаптеров.

9. Самостоятельная работа.

Самостоятельная работа студентов по курсу «Объектно-ориентированное программирование» предусматривает:

- систематическое посещение лекционных занятий, ведение конспекта лекций;
- повседневное изучение лекционного материала и содержания технической литературы, рекомендуемые этой программой и рабочим учебным планом;
- добросовестную подготовку к лабораторным занятиям;
- своевременное и качественное оформление отчётов по лабораторным работам.
- самостоятельную разработку алгоритмов и текстов программ лабораторных работ.

Темы для дополнительной самостоятельной работы:

Семестр 1

1. Использование веб-сервисов в Java SE 6: XML, SOAP, WSDL, UDDI
2. Обзор и использование технологии Facelets
3. Поддержка скриптовых языков в Java-приложениях
4. Использование Java Standart Tag Library(JSTL).
5. WEB 2.0. AJAX, GWT
6. Использование библиотеки Standart Widet Toolkit (SWT)
7. Исследование технологии Java 3D(OpenGL).
8. Сопровождение Java приложений. Архивы JAR, WAR, EAR.
9. Java ME Wireless Messaging
10. Обзор и использование технологии Struts, jsp и создание веб-приложений на Java

Семестр 2

- 1) Программное создание экрана. LayoutParams.
- 2) Метод startActivityForResult. Назначение requestCode и resultCode.
- 3) Uri. Вызов системных приложений.
- 4) Spinner – выпадающий список.
- 5) Parcel. Передача Parcelable объектов с помощью Intent.
- 6) onSaveInstanceState. Сохранение данных Activity при повороте экрана.
- 7) Handler. Отложенные сообщения, удаление из очереди, Handler.Callback.
- 8) Service. Обратная связь с помощью PendingIntent.
- 9) MultiTouch – обработка множественных касаний.
- 10) Виджеты. Создание. Жизненный цикл.

10.Индивидуальные задания.

Индивидуальные задания не предусмотрены

11. Контрольные вопросы к промежуточной аттестации

Семестр 1

1. Дайте определения полиморфизма. Какие особенности его реализации на языке Java?
2. Назовите отличия индустриального и экстремального программирования. Какие приёмы программирования существуют для двух этих подходов в языке Java
3. Опишите основные отличия класса String от StringBuffer?
4. Что такое механизм автоматической сборки мусора (garbage collector)?
5. Основные классы библиотеки для разработки тестов JUNIT
6. Какие действия необходимо произвести, чтобы создать компонент и подписаться на событие, которое он генерирует?
7. Из каких частей состоит заголовок объявления класса? Тело класса?
8. Какие элементы языка Java имеют имена? Какие из них должны быть объявлены?
9. Сравните использование операторов выбора switch в языке C# и Java.
10. Как отсортировать массив пользовательских объектов на языке Java ?
11. Когда для реализации списка типа List нужно использовать объект класса ArrayList, а когда LinkedList ?

12. Какие особенности использования коллекций для списков, множеств, пар значений?
13. Какие изменения для работы с коллекциями были внесены в версии Java 2 SE 5.0?
14. Ввести число, занести его цифры в стек. Вывести в число, у которого цифры идут в обратном порядке.
15. Задать два стека, поменять информацию местами.
16. Для каких элементов класса работает полиморфизм?
17. Нарисуйте иерархию классов ввода вывода на основе классов Reader и Writer ?
18. Как реализованы в классе Object методы equals(), toString(), hashCode()?
19. Какие объекты могут быть клонированы?
20. Какие условия должны быть выполнены при переопределении метода equals()?
21. Сколько объектов может быть создано в процессе выполнения клонирования одного объекта средствами JVM?
22. Хотя примитивные массивы не могут участвовать в преобразованиях, однако массивы int[][] и byte[][] могут рассматриваться как одномерные объектные массивы, основанные на ссылочном типе «одномерный примитивный массив». Могут ли такие типы быть преобразованы из одного в другой?
23. Как определить, можно ли преобразовать один тип массива к другому?
24. Сколько объектов порождается при инициализации массива new int[3][4]? new int[3][][]?
25. Для каких примитивных типов Java существуют классы-обертки? Что будет получено в результате выполнения: (new Integer(1)).equals(new Byte(1)) ?
26. Чем отличается внутренний (inner) класс от вложенного (nested) класса?
27. Может ли переменная иметь тип абстрактный класс? Интерфейс? Если да, то какие значения она может хранить?
28. Может ли переменная иметь тип абстрактный класс? Интерфейс? Если да, то какие значения она может хранить?
29. Как узнать, какие события генерирует стандартный графический компонент?
30. Опишите жизненный цикл потока.
31. Если объявить метод synchronized, то какой эффект будет этим достигнут?
32. Опишите механизмы синхронизации потоков
33. Как работают static synchronized методы?
34. Если один поток начал исполнение synchronized-блока, указав ссылку на некий объект, может ли другой поток обратиться к полю этого объекта? К методу?
35. Почему метод wait требует обработки InterruptedException, а методы notify и notifyAll – нет?
36. Может ли поток никогда не выйти из метода wait, даже если будет вызван метод notify? notifyAll?
37. Какие действия необходимо предпринять для установления TCP соединения между
38. Какие действия необходимо предпринять для установления TCP соединения между двумя Java-приложениями?
39. Какие действия необходимо предпринять для обмена данными по UDP протоколу?
40. Можно ли с помощью класса URL пересылать данные на сервер?
41. для чего нужны менеджеры компоновки? Исходя из каких параметров они выполняют свою работу?
42. Напишите класс-компоненту, у которого по центру рисуется квадрат размерами 10x10.
43. Может ли быть дважды вызван метод init у апплета? Метод start?

44. Опишите на память иерархию классов, применяемых для построения визуальных приложений
45. Промоделировать вращение спутника вокруг планеты по эллиптической орбите. Когда скрывается за планетой – спутник не виден.
46. Создать фрейм и разместить на нем окружность (одну или несколько). Объект должен «убегать» от указателя мыши. При приближении на некоторое расстояние объект появляется в другом месте фрейма.
47. Имея два объекта класса File, каким образом будет наиболее корректно узнать, указывают ли они на одну и ту же директорию (и на директорию ли)? Возможно ли только с помощью этих двух объектов удалить директорию? Если да, то как изменится содержимое другого объекта (если они действительно указывают на одну и ту же директорию)?
48. Какие классы предоставляют методы для записи в поток двоичного представления значений примитивных типов Java?
49. Если необходимо записать (и после считать) несколько строк в файл (из файла), в каком порядке и какие следует настроить фильтры (и для чтения, и для записи)? Какие из них можно пропустить?
50. Что произойдет при попытке к одному объекту PipedWriter присоединить несколько различных объектов PipedReader? Что произойдет, если несколько раз подряд присоединять один и тот же PipedReader?
51. Какая кодировка используется классом OutputStreamWriter по умолчанию?
52. Что будет записано в поток, если вызвать метод print() класса PrintWriter, передав в качестве параметра new File("d:\\word.txt") ?
53. Какие значения могут быть переданы в конструктор RandomAccessFile для указания режима доступа (чтение/запись)?
54. Какое значение следует передать методу seek() объекта RandomAccessFile, чтобы последний байт файла был считан одиночным вызовом read()?
55. Какие методы объявлены в интерфейсе Serializable?
56. Что произойдет, если записать в файл, используя ObjectOutputStream, значения типов long, int, byte именно в таком порядке, а считать в обратном, используя DataInputStream?

Семестр 2

1. Android SDK и Android NDK. Назначение и особенности.
2. Принципы работы с ОС Android: Activity и Intents. Определения, пример.
3. Принципы работы с ОС Android: Views, Services. Назначение, пример.
4. Принципы работы с ОС Android: ContentProvider, BroadcastReceiver.
5. Назначение.
6. Инструментарий элементов управления Android.
7. Модель обработки событий ОС Android. Пример обработчиков событий.
8. Модель документ/представление в мобильном программировании.
9. Доступ к оборудованию в ОС Android (общие принципы).
10. Пример доступа к оборудованию в ОС Android: получение снимка видекамерой.
11. Пример доступа к оборудованию в ОС Android: получение координат GPS.
12. Пример доступа к оборудованию в ОС Android: акселерометры и гироскопы.
13. Анимация и жесты в ОС Android.
14. C++ программы для ОС Android. Преимущества и недостатки.

15. Приемы для улучшения производительности и уменьшения потребления памяти для приложений Android.
16. Основные права и полномочия для запуска приложений на устройстве.
17. Работа с настройками сотовой сети, подключение голосовых услуг, получение и отправка коротких сообщений.

12. Образец экзаменационного билета

Экзаменационный билет № 1

1. Какие классы предоставляют методы для записи в поток двоичного представления значений примитивных типов Java?
2. Написать программу, для передачи текстового файла по локальной сети.
3. Требуется разработать приложение с графическим пользовательским интерфейсом, поддерживающее создание/редактирование/удаление/поиск заметок. Два варианта хранения заметок: А) В базе SQLite. Б) С использованием файловой системы.

13. Образец тестового задания

Вопрос. Что будет выведено в результате компиляции и запуска такого кода:

```
import java.awt.*;
import java.awt.event.*;
public class Quest3 extends Frame implements WindowListener{
    public Quest3(){
        setSize(300,300);
        setVisible(true);
    }
    public void windowClosing(WindowEvent e){
        System.exit(0);
    }
    public static void main(String args[]){
        Quest3 q = new Quest3 ();
    } }
```

Варианты ответов

- а) Ошибка компиляции
- б) Компиляция и запуск с выводом пустого фрейма
- в) Компиляция без запуска

г) Ошибка времени выполнения

Вопрос. Можно ли при переопределении некоторого абстрактного метода `perform()` использовать выражение `super.perform()`?

Правильный ответ нет. Выражение `super.perform()` означает полноценный вызов родительского метода, что невозможно, если у него отсутствует тело, что верно для абстрактных методов.

Вопрос. Как реализованы в классе `Object` методы `equals()`, `toString()`, `hashCode()` ?

Правильный ответ: в классе `Object` методы `equals()`, `toString()` и `hashCode()` имеют следующие реализации:

- `equals()` – возвращает `true`, если ссылки на объекты совпадают;
- `toString()` – возвращает строку, которая составляется следующим образом: название класса, символ '@', значение, возвращаемое вызовом метода;
- `hashCode()`, представленное в шестнадцатеричном виде;
- `hashCode()` – имеет native реализацию, возвращающую адрес, по которому хранится объект.

Вопросы по Android.

1. Опишите программный стек Android. Виртуальная машина Dalvik.
2. Опишите архитектуру Android приложений.
3. Опишите приемы для улучшения производительности и уменьшения потребления памяти для приложений Android
4. Опишите основные составляющие манифеста приложения.
5. Опишите жизненный цикл мобильного приложения.

14. Критерии оценивания

Шкала оценивания:

Шкала ECTS	Оценка по 100-балльной шкале, которая действует в ДонНУ	Оценка по государственной шкале (экзамен, дифференцированный зачёт)	Оценка по государственной шкале (зачёт)
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено
C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено
FX	30-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
F	0-29	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

Согласно модульному принципу организации учебного процесса содержание дисциплины «Объектно-ориентированное программирование» включает в себя четыре зачётных модуля. Каждый зачётный модуль состоит из теоретического материала и

практических задач, выполнение которых требует овладения теорией в указанном в модуле объеме.

Оценка знаний студентов проводится по 100-балльной шкале по следующим критериям:

Зачётные модули	Форма контроля	Баллы
Содержательный модуль 1	Блок лабораторных работ	26
	Контрольная работа	24
Содержательный модуль 2	Блок лабораторных работ	38
	Контрольная работа	12
	Зачет	
Общий итог за модуль 1		100
Содержательный модуль 3	Входной балл, 50% от 1 модуля	50
	Блок лабораторных работ	10
Содержательный модуль 4	Контрольная работа	24
	Блок лабораторных работ	10
	Проверка конспектов	2
	Экзамен	10
Общий итог за модуль 2		100

К первому модульному контролю студент должен защитить 6 лабораторных работ. За *первую* лабораторную работу студент может получить 2 балла. За *вторую, третью и четвертую* лабораторные работы студент может получить по 4 баллов. За *пятую и шестую* лабораторные работы студент может получить по 6 баллов.

На первом модульном контроле студент имеет возможность получить 24 балла, решив 2 практических задания. Первая задача оценивается в 12 баллов, вторая - в 12 баллов.

Ко второму модульному контролю студент должен защитить 4 лабораторные работы. За *седьмую и восьмую* лабораторные работы студент может получить по 8 баллов. За *девятую* лабораторную работу студент может получить 10 баллов. За *десятую* лабораторную работу студент может получить 12 баллов.

На втором модульном контроле студент имеет возможность получить 12 баллов, решив 2 практических задачи. Первая задача оценивается в 6 баллов, вторая – в 6 баллов.

К третьему модульному контролю студент должен защитить 4 лабораторные работы. За *одиннадцатую, двенадцатую, тринадцатую и четырнадцатую* лабораторные работы студент может получить по 6 баллов. В 2 балла оценивается ведение конспекта лекций.

На третьем модульном контроле студент имеет возможность получить 24 балла, ответив на 12 тестовых вопросов, каждый из которых оценивается в 2 балла.

К четвертому модульному контролю студент должен защитить 4 лабораторные работы. За *пятнадцатую, шестнадцатую, семнадцатую и восемнадцатую* лабораторные работы студент может получить по 6 баллов. В 2 балла оценивается ведение конспекта лекций.

На четвертом модульном контроле студент имеет возможность получить 24 балла, решив 4 практических задания. Первая задача оценивается в 4 балла, вторая - в 4 балла, третья - в 8 баллов, четвертая - в 8 баллов.

Оценка за овладение курса выставляется по следующим принципам:

– Оценку «отлично» заслуживает студент, который обнаружил глубокие знания при ответах на теоретические вопросы по темам курса, а также выполнил практические задания в полном объёме и набрал более 90 баллов.

– Оценку «хорошо» заслуживает студент, сделавший ошибки в теоретических или практических ответах, которые могут быть интерпретированы как малосущественные для вопросов, которые рассматривались. Студент должен набрать более 75 баллов.

– Оценку «удовлетворительно» заслуживает студент, который выполнил задания неполно и с ошибками, но при этом набрал более 60 баллов.

– Оценку «неудовлетворительно» заслуживает студент, который не выполнил большинства теоретических и практических задач и набрал менее 60 баллов.

15. Материально-техническое обеспечение учебного процесса

Для проведения лекционных занятий требуется аудитория на поток, оборудованная мультимедийным проектором и экраном, или интерактивной доской, или меловой доской.

Для проведения лабораторных занятий по дисциплине необходим оборудованный ПЭВМ или ноутбуками компьютерный класс с возможностью выхода в Интернет.

16. Рекомендованная литература

ОСНОВНАЯ

1. Герберт Шилдт. Java 8. Полное руководство. – Москва: Диалектика, 2015 г., 1376 с.
2. Берт Бейтс, Кэтти Сьерра. Изучаем Java. – Москва: ЭКСМО, 2012, 720 с.
3. Кей Хорстманн, Гари Корнелл. Java. Библиотека профессионала. Том 1. Основы. – Москва: Вильямс, 2014, 864 с.
4. Патрик Нимейер, Дэниэл Леук. Программирование на Java. – Москва: ЭКСМО, 2014 г., 1216 с.
5. Алексей Васильев. Java. Объектно-ориентированное программирование. – Санкт-Петербург: Питер, 2013 г., 400 с.
6. Р. Хелм, Эрик Гамма. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Санкт-Петербург: Питер, 2013, 368 с.
7. Брюс Эккель. Философия Java. Библиотека программиста. – Санкт-Петербург: Питер, 2014, 640 с.
8. Дирк Льюис, П. Мюллер, Виктор Букирев. Самоучитель Java 7. – Санкт-Петербург: BHV, 2013, 464 с.
9. Айвор Хортон. Java 2 (книга 1 и книга 2). – Москва: Лори, 2013, 1020 с.
10. Берт Бейтс, Кэтти Сьерра, Элизабет Фримен, Эрик Фримен. Паттерны проектирования. – Санкт-Петербург: Питер, 2014, 656 с.
10. Кей С. Хорстманн. Java SE 8. Вводный курс. – Москва: Вильямс, 2014, 208 с.
11. Бенджамин Эванс, Мартин Вербург. Java. Новое поколение разработки. – Санкт-Петербург: Питер, 2014, 560 с.
12. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г Буч. – М. : Вильямс, 2008. – 720 с.
13. Эккель Б. Философия Java / Б. Эккель. 4-е полное издание. — СПб.: Питер, 2015. — 1168 с.
14. Иванова Г.С. Объектно-ориентированное программирование: Учебник для студ. вузов, обуч. по направлению подгот. дипломир. специалистов "Информатика и вычисл. техника" / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев; Под ред. Г. С. Ивановой.—2-е изд., перераб. и доп.—М.:Изд-во МГТУ им. Н.Э.Баумана, 2003.

15. Хашими С., Коматинени С., Маклин Д. Разработка приложений для Android.- СПб.: Питер, 2011.–736 с.
16. Майер Р. Professional Android 2: Application Developmentecond Edition М.:Эксмо, 2011.–672 с.

ДОПОЛНИТЕЛЬНАЯ

1. Герберт Шилдт. Java. Полное руководство.– Москва: Вильямс, 2012, 1104 с.
2. Роберт Лигуори, Патриция Лигуори. Java 8. Карманный справочник. – Москва: Вильямс, 2015 г., 256 с.
3. Герберт Шилдт. Java. Руководство для начинающих. – Москва: Вильямс, 2012 г., 624 с.
4. Роберт Седжвик, Кевин Уэйн. Алгоритмы на Java. – Москва: Вильямс, 2012 г., 848 с.
5. Алексей Васильев. Самоучитель Java с примерами и программами. – Минск: Наука и техника, 2013 г., 368 с.
6. Джошуа Блох. Java. Эффективное программирование.– Москва: Лори, 2014г., 310 с.
7. Фрэд Лонг, Дхрув Мохиндра, Роберт С. Сиакорд, Дин Сазерленд, Дэвид Свобода. Руководство для программиста на Java. 75 рекомендаций по 10 написанию надежных и защищенных программ. – Москва: Вильямс, 2014 г., 256 с.
8. Вадим Монахов. Язык программирования Java и среда NetBeans. – Санкт-Петербург: БХВ-Петербург, 2012 г., 704 с. с.

17. Информационные ресурсы

1. Статьи и руководства по изучению Java - URL: <http://www.javable.com> (дата обращения: 1.08.2016)
2. Примеры использования классов для различных пакетов - URL: <http://www.exampledepot.com> (дата обращения: 1.08.2016)
3. Ресурс по языку Java - URL: <http://java.sun.com> (дата обращения: 1.08.2016)
4. JSDK-документация, которая представлена в виде html-файла URL: <http://docs.oracle.com/javase/6/docs/api/> (дата обращения: 1.08.2016)
5. Новости мира Java - URL: <http://onjava.com> (дата обращения: 1.08.2016)
6. Статьи с примерами по Java - URL: <http://www.javaspecialists.eu> (дата обращения: 1.08.2016) ;
7. Статьи с примерами по Java и исследования языка URL: <http://www.theserverside.com> (дата обращения: 1.08.2016) ;

18. Программное обеспечение

1. Операционная система Microsoft Windows.
2. Пакет jdk\jre.
3. Среды разработки JCreatorPro, NetBeans IDE, Eclipse.
4. GlassFish сервер приложений последней версии.
5. Google Android Studio 2.1 или более старших версий.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2020 год.
Протокол № 12 от «2» апреля 2020 г.
Заведующий кафедрой Ермоленко Т.В.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2021 год.
Протокол № ____ от «____» _____ 2021 г.
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2022 год.
Протокол № ____ от «____» _____ 2022 г.
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2023 год.
Протокол № ____ от «____» _____ 2023 г.
Заведующий кафедрой