

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»**
(ТУСУР)

УТВЕРЖДАЮ
Директор департамента образования
П. Е. Троян
«___» 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

Уровень образования: высшее образование - бакалавриат

Направление подготовки / специальность: 09.03.04 Программная инженерия

Направленность (профиль) / специализация: Индустриальная разработка программных продуктов

Форма обучения: заочная (в том числе с применением дистанционных образовательных технологий)

Факультет: ФДО, Факультет дистанционного обучения

Кафедра: АОИ, Кафедра автоматизации обработки информации

Курс: 3

Семестр: 6

Учебный план набора 2016 года

Распределение рабочего времени

№	Виды учебной деятельности	6 семестр	Всего	Единицы
1	Самостоятельная работа под руководством преподавателя	18	18	часов
2	Лабораторные работы	4	4	часов
3	Контроль самостоятельной работы	2	2	часов
4	Всего контактной работы	24	24	часов
5	Самостоятельная работа	188	188	часов
6	Всего (без экзамена)	212	212	часов
7	Подготовка и сдача зачета	4	4	часов
8	Общая трудоемкость	216	216	часов
			6.0	3.Е.

Контрольные работы: 6 семестр - 1

Зачет: 6 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 28.02.2018
Уникальный программный ключ:
c53e145e-8b20-45aa-a5e4dbb90e8d

Томск 2018

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 09.03.04 Программная инженерия, утвержденного 12.03.2015 года, рассмотрена и одобрена на заседании кафедры АОИ «__» 20__ года, протокол №___.

Разработчик:

Доцент Кафедра автоматизации обработки информации (АОИ) _____ Ю. В. Морозова

Заведующий обеспечивающей каф.
АОИ _____

Ю. П. Ехлаков

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФДО _____

И. П. Черкашина

Заведующий выпускающей каф.
АОИ _____

Ю. П. Ехлаков

Эксперты:

Доцент кафедры технологий электронного обучения (ТЭО) _____

Ю. В. Морозова

Доцент кафедры автоматизации обработки информации (АОИ) _____

Н. Ю. Салмина

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Основная цель дисциплины – развитие теоретических представлений и практических навыков работы с информацией, хранящейся или обрабатываемой в вычислительных системах, способом представления данных и их обработки с помощью современных информационных технологий. А также, формирование у студентов объектно-ориентированного мышления, обучение объектно-ориентированному (ОО) подходу к анализу предметной области и использованию объектно-ориентированной методологии программирования при разработке программных продуктов.

1.2. Задачи дисциплины

- изучение техники объектно-ориентированного анализа;
- изучение приемов объектно-ориентированного программирования;
- изучение технологии проектирования архитектуры информационных систем;
- изучение основ проектирования информационно-коммуникационных технологий (ИКТ) и основ управления ИКТ-проектами.
-

2. Место дисциплины в структуре ОПОП

Дисциплина «Объектно-ориентированное программирование» (Б1.В.ДВ.4.1) относится к блоку 1 (вариативная часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Информатика и программирование.

Последующими дисциплинами являются: Методы и технологии программирования, Тестирование программного обеспечения.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- ПК-1 готовностью применять основные методы и инструменты разработки программного обеспечения;

В результате изучения дисциплины обучающийся должен:

- **знать** методы обработки и способы реализации основных структур данных в объектно-ориентированных программных средах.
- **уметь** разрабатывать объектно-ориентированные программы в современных инструментальных средах.
- **владеть** практическими приемами объектно-ориентированного программирования.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 6.0 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Semestры
		6 семестр
Контактная работа (всего)	24	24
Самостоятельная работа под руководством преподавателя (СРП)	18	18
Лабораторные работы	4	4
Контроль самостоятельной работы (КСР)	2	2
Самостоятельная работа (всего)	188	188
Подготовка к контрольным работам	6	6
Оформление отчетов по лабораторным работам	8	8
Подготовка к лабораторным работам	8	8

Самостоятельное изучение тем (вопросов) теоретической части курса	166	166
Всего (без экзамена)	212	212
Подготовка и сдача зачета	4	4
Общая трудоемкость, ч	216	216
Зачетные Единицы	6.0	

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

Названия разделов дисциплины	СРП, ч	Лаб. раб., ч	КСР, ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
6 семестр						
1 Введение в методологию объектно-ориентированного программирования	4	0	2	56	60	ПК-1
2 Основные понятия объектно-ориентированного программирования.	8	4		64	76	ПК-1
3 Применение библиотек и иерархий объектов при программировании.	6	0		68	74	ПК-1
Итого за семестр	18	4	2	188	212	
Итого	18	4	2	188	212	

5.2. Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)

Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя)

Названия разделов	Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)	Трудоемкость, ч	Формируемые компетенции
6 семестр			
1 Введение в методологию объектно-ориентированного программирования	Сложность ПО. Объектная декомпозиция. Класс и объект. Типы отношений между классами и объектами. Принципы ООП.	4	ПК-1
	Итого	4	
2 Основные понятия объектно-ориентированного программирования.	Основы и синтаксис языка Java. Класс и его структура. Конструкторы. Геттеры и сеттеры. Перегрузка и переопределение методов. Вложенные и внутренние классы. Абстрактные классы.	8	ПК-1
	Итого	8	
3 Применение	Интерфейсы. Потоки. Коллекции. Обра-	6	ПК-1

библиотек и иерархий объектов при программировании.	ботка исключений. Иерархия классов.		
	Итого	6	
Итого за семестр		18	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (следующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (следующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин		
	1	2	3
Предшествующие дисциплины			
1 Информатика и программирование	+	+	
Последующие дисциплины			
1 Методы и технологии программирования	+	+	+
2 Тестирование программного обеспечения	+	+	

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Компетенции	Виды занятий				Формы контроля
	СРП	Лаб. раб.	КСР	Сам. раб.	
ПК-1	+	+	+	+	Контрольная работа, Проверка контрольных работ, Отчет по лабораторной работе, Зачет, Тест

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Таблица 7.1 – Наименование лабораторных работ

Названия разделов	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
6 семестр			
2 Основные понятия объектно-ориентированного программирования.	Лабораторная работа «Классы».	4	ПК-1
	Итого	4	
Итого за семестр		4	

8. Контроль самостоятельной работы

Виды контроля самостоятельной работы приведены в таблице 8.1.

Таблица 8.1 – Виды контроля самостоятельной работы

№	Вид контроля самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции
6 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ПК-1
Итого		2	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
6 семестр				
1 Введение в методологию объектно-ориентированного программирования	Самостоятельное изучение тем (вопросов) теоретической части курса	54	ПК-1	Зачет, Контрольная работа, Тест
	Подготовка к контрольным работам	2		
	Итого	56		
2 Основные понятия объектно-ориентированного программирования	Самостоятельное изучение тем (вопросов) теоретической части курса	54	ПК-1	Зачет, Контрольная работа, Отчет по лабораторной работе, Тест
	Подготовка к лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	2		
	Итого	64		
3 Применение библиотек и иерархий объектов при программировании	Самостоятельное изучение тем (вопросов) теоретической части курса	58	ПК-1	Зачет, Контрольная работа, Отчет по лабораторной работе, Тест
	Подготовка к лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	2		
	Итого	68		
	Выполнение контрольной работы	2	ПК-1	Контрольная работа
Итого за семестр		188		
	Подготовка и сдача зачета	4		Зачет

Итого	192		
-------	-----	--	--

10. Контроль самостоятельной работы (курсовой проект / курсовая работа)

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется.

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Морозова Ю. В. Объектно-ориентированный анализ и программирование [Электронный ресурс]: учебное пособие / Ю. В. Морозова. – Томск ФДО, ТУСУР, 2018. Доступ из личного кабинета студента. - Режим доступа: <https://study.tusur.ru/study/library/> (дата обращения: 09.08.2018).

12.2. Дополнительная литература

1. Тузовский, А. Ф. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. Издательство Юрайт, 2018. — 206 с. — (Серия Университеты России). Доступ из личного кабинета студента. - Режим доступа: <https://biblio-online.ru/book/BDEEFB2D-532D-4306-829E-5869F6BDA5F9/obektno-orientirovannoe-programmirovanie> (дата обращения: 09.08.2018).

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Морозова Ю. В. Объектно-ориентированный анализ и программирование [Электронный ресурс]: Методические указания к лабораторным работам и организации самостоятельной работы для студентов заочной формы обучения технических направлений, обучающихся с применением дистанционных образовательных технологий / Ю.В. Морозова. – Томск ФДО, ТУСУР, 2018. Доступ из личного кабинета студента. - Режим доступа: <https://study.tusur.ru/study/library/> (дата обращения: 09.08.2018).
2. Морозова Ю. В. Объектно-ориентированный анализ и программирование : электронный курс / Ю. В. Морозова. – Томск ТУСУР, ФДО, 2018. Доступ из личного кабинета студента.

12.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

12.4. Профессиональные базы данных и информационные справочные системы

1. Рекомендуется использовать информационные, справочные и нормативные базы данных <https://lib.tusur.ru/ru/resursy/bazy-dannyh> (со свободным доступом).
2. ЭБС «Лань»: [www.e.lanbook.com](http://lanbook.fdo.tusur.ru) (доступ из личного кабинета студента по ссылке <http://lanbook.fdo.tusur.ru>).

13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

13.1. Общие требования к материально-техническому и программному обеспечению дисциплины

13.1.1. Материально-техническое и программное обеспечение дисциплины

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- Google Chrome
- Kaspersky Endpoint Security 10 для Windows
- Microsoft Windows
- OpenOffice

13.1.2. Материально-техническое и программное обеспечение для лабораторных работ

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- 7-zip
- Eclipse Oxygen, Eclipse PLv2->GNU GPLv2 (с возможностью удаленного доступа)
- Google Chrome
- Java SE Development Kit (с возможностью удаленного доступа)
- Microsoft Windows
- OpenOffice

13.1.3. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:

- учебная мебель;

- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

1. Как называется каталог для проектов в IDE, в котором располагаются файлы проекта?

- Perspective
- Console
- Workspace
- Views

2. Какой пакет необходимо импортировать в проект для арифметических операций произвольной точности?

- java.math
- java.util
- java.lang
- java.io

3. Как называется подсказка в IDE, вызываемая командой контекстного меню Java-редактора и предлагает различные варианты исправления ошибки?

- Quick Fix
- Content Assist
- Console
- Javadoc

4. В каком файле содержится байт-код после компиляции программы, написанной на языке Java?

- в файле с расширением .class
- в файле с расширением .java
- в файле с расширением .exe
- в файле с расширением .bin

5. Как правильно описывается метод main()?

- public static void main(String[] args) { /* тело метода */ }
- int main(String [] args) { /* тело метода */ }
- public void main(String[] args) { /* тело метода */ }
- void main(String[] args) { /* тело метода */ }

6. В каком представлении отображаются проблемы сборки проекта?

- Problems
- Concole
- Declaration
- Javadoc

7. Какой пакет является наиболее важным из всех пакетов, входящих в Java API, поскольку включает классы, составляющие основу для всех других классов?

- java.lang
- java.io
- java.util
- java.awt

8. Что произойдет при выполнение следующего кода программы?

```
int x=3;
int y=5;
System.out.println (x/y);
```

- 0
- 1
- 0.6
- ошибка компиляции

9. Каким будет результат компиляции и выполнения следующего кода?

```
import java.util.*;
import java.util.HashSet;
import java.util.Set;
public class Main {
    public static void main(String args[]){
        Set<Number> numbers = new HashSet<Number>();
        numbers.add(4);
        numbers.add(new Integer(4));
        numbers.add(new Long(4));
        System.out.println(numbers);
    }
}
```

- [4, 4]
- [4]
- [4, 4, 4]
- ошибка компиляции

10. Что произойдёт в результате компиляции и выполнения данного кода?

```
public class Main {
    public void method() {
        String a = "1";
        int b = 2;
        System.out.println(a + b);
    }
    public static void main(String[] args)
    {
```

```
method();
}
}
```

- Нельзя вызвать метод method() без создание объекта класса Main, так как он нестатический.

- 12
- 3
- 1+2

11. Какое ключевое слово используется, чтобы указать, что класс реализует интерфейс?

- implements
- extends
- throws
- default

12. Какой принцип ООП, постулирующий возможность замены объектов со сходным интерфейсом?

- Полиморфизм
- Абстракция
- Инкапсуляция
- Наследование

13. При создании объекта вначале объявляется переменная класса, а затем с помощью ключевого слова и конструктора класса непосредственно создается объект, на который и будет указывать объявленная эта переменная. Какое ключевое слово необходимо указать при создании объекта?

- new
- this
- super
- instanceof

14. Какие методы не имеют доступа к данным объекта, и для их использования не нужно создавать экземпляры (данного класса)?

- static
- abstract
- final
- strictfp

15. Какой модификатор класса означает невозможность наследования от этого класса?

- static
- abstract
- final
- strictfp

16. С помощью какого ключевого слова из метода дочернего класса можно вызвать переопределенный метод родительского класса?

- super
- this
- instanceof
- new

17. С помощью инкапсуляции можно скрыть (ограничить доступ) к важным членам данных в своем коде, что улучшит безопасность. Какие для этого необходимо использовать модификаторы?

- Модификаторы public, private, protected
- Модификатор static
- Модификаторы final
- Модификаторы strictfp и native

18. Каким будет результат компиляции и выполнения следующего кода?

```
String s1 = new String("hello");
```

```
String s2 = "hello";
System.out.println(s1==s2);
```

- false
- true
- Ошибка компиляции
- Ошибка выполнения

19. Каким будет результат компиляции и выполнения следующего кода?

```
int[] numbers = {-9, 6, 0, -59};
System.out.println(numbers.length());
```

- Ошибка компиляции
- Ошибка выполнения
- 4
- numbers.length()

20. Каким будет результат компиляции и выполнения следующего кода в одном пакете проекта?

```
class Demo {
private int a = 5;
public int b = 6;
int c = 7;
}
```

```
public class Main {
public static void main(String[] args) {
Demo f = new Demo();
System.out.print(" " + f.a);
System.out.print(" " + f.b);
System.out.print(" " + f.c);
}
}
```

- Ошибка компиляции
- Ошибка выполнения
- 5 6 7
- 6 7

14.1.2. Темы контрольных работ

Объектно-ориентированное программирование

1. Что будет напечатано?

```
System.out.println("8/2=" + 8/2 == "8/2=4");
```

- 0
- ошибка компиляции
- false
- true

2. Какие из следующих строк скомпилируются без ошибки?

```
float f = 7.0;
int i = 32565;
boolean n = null;
byte b = 255;
```

3. Что будет на консоле?

```
int i=11;
System.out.print(i%2==1);
2
true
```

false
1
ошибка компиляции

4. Сколько раз выполнится цикл?

```
int i = 1;  
  
do {  
  
    System.out.print(i + " итерация");  
  
    i++;  
  
}  
  
while (i < 5);
```

5. Что будет напечатано в результате выполнения данного кода?

```
int i = 1;  
do while (i < 1)  
System.out.println("i = " + ++i);  
while (i > 1);
```

i=1
Программа выполнится, но ничего не выведет на экран
i=2

6. Что будет напечатано?

```
int b[] = new int[5];  
  
for (int i = 1; i <= b.length; i++) {  
  
    b[i] = i + 1;  
  
    System.out.print(b[i]);  
  
}
```

Ошибка выполнения

1 2 3 4 5

Ошибка компиляции

7. Что будет выведено на консоль?

```
String s1 = new String("hello");  
String s2 = new String("Hello");  
System.out.println(s1.equals(s2) == s2.equals(s1));
```

8. Каким будет результат выполнения данного кода?

```
String s1 = "Hello";
```

```
String s2 = new String("Hello");
if (s1 == s2) {
    System.out.println("True");
} else { System.out.println("False"); }
```

9. Как называется принцип ООП, постулирующий возможность замены объектов со сходным интерфейсом?

- агрегация
- инкапсуляция
- композиция
- абстракция
- полиморфизм

10. Какое ключевое слово используется, чтобы указать, что класс реализует интерфейс?

- throws
- interface
- extends
- implements

14.1.3. Зачёт

Приведены примеры типовых заданий из банка контрольных тестов, составленных по прошальным разделам дисциплины.

1. Что будет в результате выполнения следующего кода программы при f(59)?

```
public static void f(int x){
    int x1, x0;
    if(x<10) {
        x0 = -1;
        x1 = -1;
        System.out.print(x1, x0);
    }
    if(x>99){
        x0 = 1;
        x1 = 1;
        System.out.print(x1, x0);
    }
    x0 = x%10;
    x1 = (x-x0)/10
    System.out.print(x1);
    System.out.print(x0);
}
```

Ведите ответ в текстовое поле.

2. Сколько раз выполняются циклы в результате выполнения следующего кода программы при f(24)?

```
public static void f(int n){
    int x, x1, i, j;
    x = 0;
    x1 = 1;
    i = 0;
    j = 0;
    while(x < n){
        x += 1;
        i++;
    }
}
```

```
do{  
    x1 += 2;  
    j++;  
}while(x1 < n);  
System.out.print(i);  
System.out.print(j);  
}  
Введите ответ в текстовое поле.
```

3. Что будет результатом выполнения?

```
class Catty {  
  
    Catty(int ii) {  
        System.out.println("age of a cat = " + ii);  
    }  
}
```

```
public class Cat {  
    private final int i = 0;  
  
    private final int j;  
  
    private final Catty p;
```

```
    public Cat() {  
        j = 1;  
        p = new Catty(j);  
    }
```

```
    public Cat(int x) {  
        j = x+1;  
        p = new Catty(j);  
    }
```

```
    public static void main(String[] args) {  
        new Cat();  
        new Cat(g);  
    }
```

- age of a cat = 1
- age of a cat = 48
- ошибка компиляции
- ошибка выполнения, так как метод main() статический.

4. Что будет в результате выполнения?

```
class Catty {  
    Catty(int i) {  
        System.out.println("age of a cat = " + i);  
    }  
}
```

```
public class Cat {  
    public static void main(String[] args) {
```

```
new Catty(j);
}
}
• age of a cat = j
• ошибка компиляции
• ошибка выполнения, так как метод main() статический.
```

5. Что будет в результате выполнения?

```
public class Cat {
public Cat(int age) {
System.out.println("Cat() = "+age);
}

public void Cat(int age) {
System.out.println("void Cat() = "+age);
}

void method(int age) {
System.out.println("void method() = "+age);
}

public static void main(String[] a) {
Cat tom = new Cat(j);
tom.method(5);
tom.Cat(6);
}
```

• Cat() = 5

void method() = 5

void Cat() = 6

• ошибка компиляции

• ошибка выполнения, так как Cat() не может быть void.

6. Что будет в результате выполнения?

```
class A
```

```
{
```

```
final public int GetResult(int a, int b) { return 0; }
```

```
}
```

```
class B extends A
```

```
{
```

```
public int GetResult(int a, int b) {return 1; }
```

```
}
```

```
public class Cat
```

```
{
```

```
public static void main(String args[])
{
    B b = new B();
    System.out.println("x = " + b.GetResult(0, 5));
}
```

- x = 5
- ошибка компиляции
- x = 0

7

Что будет в результате выполнения?

```
public class Cat
{
    public static void main(String args[])
    {
        Float cats = new Float("10");
        switch (cats)
        {
            case 10: System.out.println("Ten");
            case 0: System.out.println("Zero");
            default: System.out.println("Default");
        }
    }
}
```

- Ten
- ошибка компиляции
- ошибка выполнения
- Zero
- Default

8. Что будет результатом компиляции и выполнения данного кода:

```
import java.util.ArrayList;
```

```
public class Test {  
    public static void main(String[] args) {  
        ArrayList<Integer> list = new ArrayList<>(2);  
        list.add(1);  
        list.add(2);  
        list.add(3);  
        System.out.println(list.size());  
    }  
}
```

3

2

Ошибка компиляции.

9. Выберите метод класса Object, который можно переопределять при наследовании.

sort()
println()
add()
toString()

10. Какой абстрактный класс лежит в основе иерархии классов символьных потоков записи?

Reader
Writer
OutputStream
InputStream

11. Экземпляры какого класса позволяют взаимодействовать с объектами файловой системы?

FileOutputStream
FileInputStream
System
File

12. Базовый класс, определяющий потоковый байтовый вывод?

Writer
Reader
InputStream
FileOutputStream
OutputStream

13. Каким будет результат выполнения программы?

```
public class Test {  
    static int sum(int a, int b) {  
        try { return a + b; }  
        finally { return 0; }  
    }  
  
    public static void main(String args[]) {  
        System.out.print(sum(1, 2));  
    }  
}
```

0

30

3

Произойдёт ошибка компиляции.

14. Каков результат работы программы ниже?

```
public class Demo {  
  
    private static String test() {  
        try {  
            String str = null;  
            return str.toString();  
  
        } finally {  
            return "hello finally";  
        }  
    }  
  
    public static void main(String[] args) {  
  
        System.out.println(test());  
    }  
}  
  
hello finally  
hello finally  
NullPointerException
```

15. С помощью какого ключевого слова можно принудительно выбросить исключение?

finally
throw
exception
throws
try catch

16. Какие блоки в конструкции try/catch/finally могут отсутствовать?

try
finally
catch

17. Каким будет результат компиляции и выполнения следующего кода:

```
class Catty{  
int x =0;  
public void printX(){  
System.out.print(x);  
}  
}  
  
class Kitten extends Catty{  
int x = -1;  
@Override  
public void printX(){  
System.out.print(x);  
}
```

```
    }
    public class Cat {
    public static void main(String[] args) {
        Catty a=new Kitten();
        a.printX();
    }
    }
0
1
-1
```

Произойдёт ошибка выполнения

```
18. public class CatDemo extends Kitten {
    public static String sing() { return "Catty"; }
```

```
public static void main(String[] args) {
    CatDemo t = new CatDemo();
    Kitten s = new CatDemo();
    System.out.println(t.sing() + " " + s.sing());
}
}
```

```
class Kitten { public static String sing() { return "Kitten"; } }
```

Результат выполнения программы:

Catty Kitten

Catty Catty

Kitten Catty

```
19. class Toy {
    public void printName() {
        System.out.println("Toy");
    }
}
```

```
class Doll extends Toy {
    public void printName() {
        System.out.println("Doll");
    }
}
```

```
public class Demo {
    public static void main(String[] args) {
        Toy t = new Doll();
        t.printName();
    }
}
```

Результат выполнения программы:

Doll Toy

Toy

Doll

20. Какой модификатор поля означает его принадлежность контексту класса, а не объекта?

abstract

static

14.1.4. Темы лабораторных работ

Лабораторная работа «Классы».

14.1.5. Методические рекомендации

Учебный материал излагается в форме, предполагающей самостоятельное мышление студентов, самообразование. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Начать изучение дисциплины необходимо со знакомства с рабочей программой, списком учебно-методического и программного обеспечения. Самостоятельная работа студента включает работу с учебными материалами, выполнение контрольных мероприятий, предусмотренных учебным планом.

В процессе изучения дисциплины для лучшего освоения материала необходимо регулярно обращаться к рекомендуемой литературе и источникам, указанным в учебных материалах; пользоваться через кабинет студента на сайте Университета образовательными ресурсами электронно-библиотечной системы, а также общедоступными интернет-порталами, содержащими научно-популярные и специализированные материалы, посвященные различным аспектам учебной дисциплины.

При самостоятельном изучении тем следуйте рекомендациям:

- чтение или просмотр материала необходимо осуществлять медленно, выделяя основные идеи; на основании изученного составить тезисы. Освоив материал, попытаться соотнести теорию с примерами из практики;
- если в тексте встречаются термины, следует выяснить их значение для понимания дальнейшего материала;
- необходимо осмысливать прочитанное и изученное, отвечать на предложенные вопросы.

Студенты могут получать индивидуальные консультации с использованием средств телекоммуникации.

По дисциплине могут проводиться дополнительные занятия в форме вебинаров. Расписание вебинаров публикуется в кабинете студента на сайте Университета. Запись вебинара публикуется в электронном курсе по дисциплине.

14.2. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения
С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами исходя из состояния обучающегося на момент проверки

14.3. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- представление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.