Документ подписан простой электронной подписью
Информация о владельце: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 06.07.2026 15:38:44
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информационные технологии»

УТВЕРЖДАЮ
Декан факультета
«Информационных технологий»
_____ / Д.Г. Демидов /
« 26 » _____февраля 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Прикладное программирование»

Направление подготовки
09.03.03 Прикладная информатика

Профиль
«Разработка и интеграция бизнес-приложений»

Квалификация
Бакалавр

Формы обучения
очная
Год начала обучения – 2026 г.

Москва, 2026 г.

Разработчик(и):



ст.преподаватель /Е.А.Харченко/

Согласовано:
Заведующий кафедрой «Инфокогнитивные технологии»,



доцент, к.т.н. /Е.А.Пухова/

Руководитель образовательной программы



ст преподаватель / М.В. Даньшина /

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 4
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 5
3.3 Содержание дисциплины 7
3.4 Тематика семинарских/практических и лабораторных занятий 9
3.5 Тематика курсовых проектов (курсовых работ) 9
4 Учебно-методическое и информационное обеспечение 11
4.1 Нормативные документы и ГОСТы 11
4.2 Основная литература 11
4.3 Дополнительная литература 11
4.4 Электронные образовательные ресурсы 12
4.5 Лицензионное и свободно распространяемое программное обеспечение 12
4.6 Современные профессиональные базы данных и информационные справочные системы 12
5 Материально-техническое обеспечение 12
6 Методические рекомендации 12
6.1 Методические рекомендации для преподавателя по организации обучения 12
6.2 Методические указания для обучающихся по освоению дисциплины 12
7 Фонд оценочных средств 13
7.1 Методы контроля и оценивания результатов обучения 13
7.2 Шкала и критерии оценивания результатов обучения 14
7.3 Оценочные средства 18

1 Цели, задачи и планируемые результаты обучения по дисциплине

К основным целям освоения дисциплины относится:

- получение знаний об основных конструкциях и идиомах языка программирования Java;
- умение на практике написать программу для выполнения поставленной аналитической задачи;
- закрепление получаемых в семестре знаний и навыков на практике; формирование взаимосвязей, получаемых в семестре знаний и
- навыков с изученными ранее и изучаемых параллельно с данной дисциплиной; подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра.

К основным задачам дисциплины относятся:

- освоение методологии, анализа и выбора принципов и методов программирования прикладных задач на языке программирования высокого уровня;
- изучение и освоение теоретического материала, как в процессе контактной, так и в ходе самостоятельной работы;
- выполнение предоставленных практических заданий различных форм, как в процессе контактной, так и в ходе самостоятельной работы;
- самостоятельная работа над тематикой дисциплины для формирования компетенций основной образовательной программы (далее, ООП).

Обучение по дисциплине «Прикладное программирование» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ИОПК-7.1. Знает основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ИОПК-7.2. Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ИОПК-7.3. Владеет языком программирования, методами отладки и тестирования работоспособности программы

2 Место дисциплины в структуре образовательной программы

Дисциплина «Прикладное программирование» относится к обязательной части (части, формируемой участниками образовательных отношений) блока Б1 «Дисциплины (модули)».

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Алгоритмы и структуры данных;
- Основы программирования;

- Разработка мобильных приложений;
- Проектирование баз данных;
- Методология программирования.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет __3__ зачетных(е) единиц(ы) (_108_ часов).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			2 семестр	
1	**Аудиторные занятия**	**54**	54	
	В том числе:			
1.1	Лекции	8	8	
1.2	Семинарские/практические занятия	-	-	
1.3	Лабораторные занятия	46	46	
2	**Самостоятельная работа**	**54**	54	
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен		экзамен	
	Итого:	**108**	108	

3.1.2 Очно-заочная форма обучения

Не проводится

3.1.3 Заочная форма обучения

Не проводится

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение. Среда разработки	8	4				4
2	UML-диаграмма классов Проект «Birds»	4	2				2
3	Лабораторная работа 1. Документирование посредством UML	4			2		2
4	Лабораторная работа 2. Проектирование посредством UML	4			2		2
5	Контейнеры стека.	4	2				2

	Контейнеры: односторонняя очередь. Контейнеры: двухсторонняя очередь						
6	Лабораторная работа 3. Применение стека	4			2		2
7	Лабораторная работа 4. Реализация дека.	4			2		2
8	Лабораторная работа 5. Применение дека	4			2		2
9	Контейнеры: односвязный и двусвязный список Лабораторная работа 6. Применение односвязного списка	4			2		2
10	Лабораторная работа 7. Реализация двусвязного списка	4			2		2
11	Шаблоны методов и классов. Лабораторная работа 8. Использование шаблона класса	4			2		2
12	Лабораторная работа 9. Разработка шаблона класса	4			2		2
13	Регулярные выражения. Лабораторная работа 10. Формирование регулярных выражений	4			2		2
14	Лабораторная работа 11. Модификация проекта «Рекурсивный компилятор формул»	4			2		2
15	Лабораторная работа 12. Модификация проекта «Стековый компилятор формул»	4			2		2
16	Лабораторная работа 13. Модификация проекта «Стековый калькулятор»	4			2		2
17	Модульное тестирование Лабораторная работа 14. Разработка модульных тестов	4			2		2
18	Лабораторная работа 15. Модификация проекта «Выпуклая оболочка»	4			2		2
19	Лабораторная работа 16. Модификация проекта «Выпуклая оболочка»	4			2		2
20	Лабораторная работа 17. Модификация проекта «Изображение полиэдра»	4			2		2
21	Шаблон проектирования MVC. Обработка событий. Проект «Счетчик» Лабораторная работа 18. Модификация GUI-приложения «Счетчик»	8			4		4
22	Лабораторная работа 19. Модификация GUI-приложения «Трехмерная модель куба»	8			4		4

23	Лабораторная работа 20. Модификация GUI-приложения базы данных «Сотрудники»	4			2		2
24	Лабораторная работа 21. Модификация проекта «Изображение полиэдра»	4			2		2
Итого		**108**	**8**		**46**		**54**

3.2.2 Очно-заочная форма обучения

Не проводится

3.3.3 Заочная форма обучения

Не проводится

3.3 Содержание дисциплины

1. Введение

Предмет, задачи и содержание дисциплины. Структура курса, его место и роль в подготовке специалиста, связь с другими дисциплинами. Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах). Особенности языка Java: интерпретируемость, независимость от платформы, мобильность. Виртуальная машина Java. JIT-компиляция.

2. Среда разработки

15Средства разработки Java-приложений. Интегрированные среды разработки. Пример с16оздания простейшей программы на языке Java в среде NetBeans, её компиляция в байт-код и запуск. Базовые пакеты и классы Java. Проекты. Пакеты. Уровни видимости классов. Импорт классов. Сборка мусора в языке Java. Структура проекта NetBeans. Создание в NetBeans простейшего приложения Java. Компиляция файлов проекта и запуск приложения. Jar-файлы. Отладка приложений в среде NetBeans. Документирование исходного кода в Java.

3. Основы ввода/вывода в языке программирования Java

Основные группы классов и интерфейсов пакета java.io. Стандартные потоки ввода-вывода. Организация ввода и вывода данных. Класс Scanner. Ввод/вывод в Java. Фильтрованные потоки. Ввод/вывод в Java. Буферизированные потоки. Ввод/вывод в Java. Канальные потоки. Ввод/вывод в Java. Синхронизация потоков данных. Работа с файлами и папками с помощью объектов типа File.

4. Типы данных языка программирования Java

Идентификаторы. Переменные и типы. Примитивные и ссылочные типы. Встроенные типы данных. Способы задания литералов различных типов. Управляющие последовательности. Символы Unicode. Специальные символы. Булевый (логический) тип. Основные операторы для работы с логическими величинами. Целые типы, переменные, константы. Основные операторы для работы с целочисленными величинами. Вещественные типы и класс Math. Работа со ссылочными переменными.

5. Операторы языка программирования Java

Оператор присваивания. Порядок действий (приоритет операторов). Приведение типов. Неявное приведение типов. Явное приведение, автоматическое расширение типов и instanceof. Строковое приведение. Правила явного и автоматического преобразования типа при работе с числовыми величинами. Метки в языке Java. Управляющие инструкции. Операторы и блоки.

Составной оператор. Условный оператор if. Оператор выбора switch. Условное выражение ...?... :

Операторы инкремента ++ и декремента -- . Оператор цикла for. Ошибки при использовании вещественного счетчика цикла. Эффективная организация циклов при вычислениях в формате с плавающей точкой. Особенности целочисленных вычислений. Организация циклов, приоритет операторов и арифметическое переполнение. Цикл с предусловием. Оператор цикла while. Цикл с постусловием. Оператор цикла do...while. Операторы прерывания continue, break, return, System.exit. Арифметические операторы. Логические операторы. Операторы сравнения. Побитовые операторы. Создание и работа с массивами. Одномерные массивы. Многомерные массивы. Присваивание и сравнение массивов. Коллекции, списки, итераторы. Перебор в цикле элементов коллекций. Оператор цикла for-each.

6. Основы объектно-ориентированного программирования в языке программирования Java

Сравнение парадигм процедурного программирования и объектно-ориентированного программирования. Основополагающие принципы объектно-ориентированного программирования: инкапсуляция, наследование и полиморфизм. Средства реализации инкапсуляции в языке Java. Классы языка программирования Java. Методы и поля. Конструкторы. Конструктор по умолчанию. Зарезервированные слова super и this. Блоки инициализации. Конструкторы в наследуемых классах. Порядок вызова конструкторов. Удаление неиспользуемых объектов и метод finalize. Проблема деструкторов для сложно устроенных объектов. Оболочечные классы. Упаковка (boxing) и распаковка (unboxing). Модификаторы уровня доступа default, public, protected, private. Управление наследованием в языке Java. Наследование. Суперклассы и подклассы. Переопределение методов. Наследование и правила видимости. Зарезервированное слово super. Проблемы множественного наследования классов. Интерфейсы. Отличия интерфейсов от классов. Проблемы наследования интерфейсов. Композиция как альтернатива множественному наследованию. Интерфейсы как средство реализации множественного наследования. Средства реализации полиморфизма в языке Java. Правила совместимости ссылочных типов как основа использования полиморфного кода. Приведение и проверка типов. Методы. Значения параметров. Применение методов для ограничения доступа. Сигнатура метода. Перегрузка методов. Статическое и динамическое связывание методов. Полиморфизм. Иерархия классов Java. Коренной класс Object и его методы. Функции. Модификаторы. Передача примитивных типов в функции. Локальные и глобальные переменные. Модификаторы доступа и правила видимости. Ссылка this. Передача ссылочных типов в функции. Проблема изменения ссылки внутри подпрограммы. Статические (static) вложенные классы и интерфейсы. Внутренние (inner) классы. Локальные (local) классы. Анонимные (anonimous) классы и обработчики событий. Анонимные (anonimous) классы и слушатели событий (listeners). Статические члены класса. Абстрактные классы и методы.

7. Профилирование Java-приложений

Принцип «презумпции виновности» и тестирование классов языка Java. Профилирование Java-приложений. Настройка профилировщика «на лету». Точки профилирования. Профилирование использования памяти классами Java.

8. Исключительные ситуации языка программирования Java

Исключительные ситуации. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Контролируемые и неконтролируемые исключения. Объявление типа исключительной ситуации и оператор throw. Объявление метода, который может

возбуждать исключительную ситуацию. Зарезервированное слово throws. Создание новых типов исключений. Оператор throw. Условие throws. Операторы try, catch и finally.

9. События языка программирования Java

Понятие события. Типы событий. Иерархия классов событий. События в Java. Модель делегирования событий. Интерфейсы блоков прослушивания событий. Способы реализации блока прослушивания. Использование внутренних классов, классов-адаптеров для упрощения обработки событий. Типы-перечисления (enum).

10. Работа с классами языка программирования Java

Работа с датами и временем. Класс GregorianCalendar. Работа со строками в Java. Строки как объекты. Классы String, StringBuffer и StringBuilder. Операции со строками. Сравнение строк. Поиск и извлечение строк. Посимвольная обработка строк. Пословное реверсирование строки. Преобразование всех символов строки к верхнему или нижнему регистру. Классы для работы с текстом. Поиск по образцу с помощью регулярных выражений. Специальные символы регулярных выражений. Классы Java Matcher и Pattern, используемые для работы с регулярными выражениями. Нахождение соответствий в тексте с помощью регулярных выражений. Нахождение всех вхождений по образцу. Замена текста, отвечающего соответствию.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

Не запланировано учебным планом

.

3.4.2 Лабораторные занятия

ЛР-1 Документирование посредством UML

Цель выполнения лабораторной работы: дать студентам практический навык в изучении программного кода в соответствии с диаграммой классов.

Результат: Измененный программный код в соответствии с предложенной диаграммой классов

ЛР-2 Проектирование посредством UML

Цель выполнения лабораторной работы: дать студентам практический навык в использовании UML

Результат: С помощью приложения UMLet построенные диаграммы классов.

ЛР-3 Применение стека.

Цель выполнения лабораторной работы: дать студентам практический навык в написании программы, которая с помощью стека проверяет правильность расстановки парных скобок в ввенной строке.

ЛР-4 Реализация дека.

Цель выполнения лабораторной работы: дать студентам практический навык в создании непрерывной реализации ограниченной двусторонней очереди целых чисел на базе вектора.

Результат: созданный дек из пяти элементов, который последовательно выводит из него элементы в прямом и обратном направлениях (без модификации класса Deq).

ЛР-5 Применение дека.

Цель выполнения лабораторной работы: Дать студентам практический навык в написании программы, которая с помощью дека определяет, является ли введенная строка палиндромом.

Результат: написанная программа.

ЛР-6 Применение односвязного списка.

Цель выполнения лабораторной работы: дать студентам практический навык в написании программ, вводящую с клавиатуры (или из файла) последовательность действий пользователя над приложениями (запуск приложения, переключение между приложениями, закрытие приложения) и выводящую список приложений в том порядке, в котором с ними работал пользователь.

Результат: Написанная работающая программа.

ЛР-7 Реализация двусвязного списка.

Цель выполнения лабораторной работы: дать студентам практический навык в построении ссылочной реализации ограниченного двусвязного списка целых чисел на основе трех одномерных статических массивов и создании двусвязного списка из пяти элементов и последовательно выводящий его элементы в прямом и обратном направлениях.

Результат: Написанная рабочая программа.

ЛР-8 Использование шаблона класса.

Цель выполнения лабораторной работы: познакомить студентом на практике с шаблонами классов.

Результат: **Написанная программа.**

ЛР-9 Разработка шаблона класса.

Цель выполнения лабораторной работы: познакомить студентов с **методами дополнения шаблона классов.**

Результат: Разработанный программный продукт.

ЛР-10 Формирование регулярных выражений.

Цель выполнения лабораторной работы: получить практические навыки в составлении регулярного выражения, задающее язык корректных адресов электронной почты.

Результат: Программный продукт, реализованный в соответствии с заданием.

ЛР-11. Модификация проекта «Рекурсивный компилятор формул».

Цель выполнения лабораторной работы: получить практические навыки в работе с арифметическими операциями и формулами, многосимвольными идентификаторами.

ЛР-12. Модификация проекта «Стековый компилятор формул».

ЛР-13. Модификация проекта «Стековый калькулятор».

Цель выполнения лабораторной работы: получить практические навыки в написании программы, которая реализует формулы с многозначными числами.

ЛР-14. Разработка модульных тестов.

Цель выполнения лабораторной работы: получить практические навыки в разработке модульных тестов.

ЛР-15. Модификация проекта «Выпуклая оболочка».

Цель выполнения лабораторной работы: получить практические навыки в написании программы.

ЛР-16. Модификация проекта «Выпуклая оболочка».

Цель выполнения лабораторной работы: получить практические навыки в написании программы.

ЛР-17. Модификация GUI-приложения «Счетчик».

ЛР-18. Модификация GUI-приложения «Трехмерная модель куба».

ЛР-19. Модификация GUI-приложения базы данных «Сотрудники».

ЛР-20. Модификация проекта «Изображение полиэдра»

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

2. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика, утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922.

3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры;

4. Порядок проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры, утвержденный приказом Минобрнауки России от 29 июня 2015 г. № 636;

5. Положение о практической подготовке обучающихся, утвержденное приказом Министерства науки и высшего образования Российской Федерации и Министерства просвещения Российской Федерации от 5 августа 2020 г. № 885/390;

6. Устав и локальные нормативные акты Московского политеха

Области профессиональной деятельности и сферы профессиональной деятельности, в которых выпускники, освоившие программу бакалавриата (далее - выпускники), могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом)

Выпускники могут осуществлять профессиональную деятельность в других областях и (или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника, предъявляемым соответствующими профессиональными стандартами.

4.2 Основная литература

1. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2023. — 219 с. — (Высшее образование). — ISBN 978-5-9916-9983-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/511703

2. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для среднего профессионального образования / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2023. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-14744-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/519727

4.3 Дополнительная литература

1. Вязовик, Н.А. Программирование на Java [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 603 с. — Режим доступа: https://e.lanbook.com/book/100405. — Загл. с экрана.

2. Лафоре Р., «Структуры данных и алгоритмы в Java» Издательство Питер, Серия Классика Computer Science, 2013 – ISBN 978-5-4461-0939-5

3. Эккель Б., «Философия Java»https://lyapidov.ru/bruce-eckel-thinking-in-java-4th-edition/

4. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2022. — 206 с. —

(Высшее образование). — ISBN 978-5-534-00849-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/490369

5. Е.А. Роганов. Основы информатики и программирования: Учебное пособие – М.: МГИУ, 2001. – 315 с.

4.4 Электронные образовательные ресурсы

1. Курс Прикладное программирование https://lms.mospolytech.ru/course/view.php?id=6530

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Microsoft Windows.
2. Язык программирования Java версии 8.1.
3. Интегрированная среда разработки IDE NetBeans 8.1.
4. Git.
5. Microsoft Office 365

4.6 Современные профессиональные базы данных и информационные справочные системы

1. Федеральная государственная информационная система - Национальная электронная библиотека (НЭБ) https://нэб.рф

5 Материально-техническое обеспечение

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенной к нему электронной доской.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения для лекций, задачи для практических работ и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

При проверке работ и отчетов следует учитывать не только правильность выполнения заданий практических работ, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия и лекции, материалы лабораторных работ.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста в области Веб-технологий.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- Опросы разделов
- Контрольные вопросы разделов
- Тестирование (итоговый тест)
- Подготовка к Лабораторным работам и их защита
- Чтения литературы и освоения дополнительного материала в рамках тематики дисциплины
- Подготовка к текущей аттестации
- Подготовка к промежуточной аттестации

Образцы заданий для проведения текущего контроля, банка тестовых заданий приведены в Разделе 7.3.1.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой дисциплины.

Образцы вопросов для проведения промежуточных аттестаций приведены в Разделе 7.3.2.

7.2 Шкала и критерии оценивания результатов обучения

Балльно-рейтинговая система, включающая следующие критерии оценки.

Критерии	Значение критерия
Выполнение и защита лабораторных работ в срок (для лабораторных работ 1-9)	+5 баллов за каждую защищенную на отлично лабораторную работу; +4 балла за каждую защищенную на хорошо лабораторную работу; +3 балла за каждую защищенную на удовлетворительно лабораторную работу; Максимальное значение критерия – не более 45 баллов.
Выполнение и защита лабораторных работ в срок (для лабораторной работы 10)	+15 баллов за каждую защищенную на отлично лабораторную работу; +10 балла за каждую защищенную на хорошо лабораторную работу; +5 балла за каждую защищенную на удовлетворительно лабораторную работу; Максимальное значение критерия – не более 15 баллов.
Невыполнение и/или не защита (защита с оценкой «неудовлетворительно») лабораторных работ в срок	-5 баллов за каждую лабораторную работу;
Выполнение экзаменационного задания	Максимальное значение критерия – 40 баллов.

Максимальная сумма набираемых по дисциплине баллов – 100. С началом каждого нового семестра изучения дисциплины набранные баллы обнуляются и рейтинг студента ведется заново. Перевод набранных баллов в оценку промежуточной аттестации производится согласно следующей таблице.

Оценка по балльно-рейтинговой системе	Оценка по итоговой аттестации
0 … 49	Неудовлетворительно
50 … 69	Удовлетворительно
70 … 84	Хорошо
85 … 100	Отлично

Шкалы оценивания результатов лабораторных работ, курсовых работ, курсовых проектов

Шкала оценивания	Описание
Отлично	Задание выполнено полностью и в срок. Отсутствуют ошибки в полученном результате. При процедуре защиты студент уверенно отвечает на контрольные вопросы, оперирует приобретенными знаниями и умениями, объясняет все этапы получения результата, его характеристики и причины их значений. Способен при необходимости доработать полученные результаты в соответствии с любыми незначительными изменениями в задании.
Хорошо	Задание выполнено полностью и в срок. Присутствуют незначительные ошибки в полученном результате. При процедуре защиты студент правильно отвечает на вопросы о ходе работы, оперирует приобретенными знаниями и умениями, однако возможны незначительные ошибки на дополнительные вопросы, в том числе и на вопросы для самоконтроля. Студент объясняет все этапы получения результата, его характеристики и причины их значений. Способен при необходимости доработать полученные результаты в соответствии с большинством незначительных изменений в задании.
Удовлетворительно	Задание выполнено либо со значительными ошибками, либо с опозданием. При процедуре защиты студент некорректно отвечает на некоторые дополнительные вопросы, в том числе и на вопросы для самоконтроля. Студент объясняет все этапы получения результата, его характеристики и причины их значений. Способен при необходимости доработать полученные результаты в соответствии с лишь некоторыми незначительными изменениями в задании.
Неудовлетворительно	Задание полностью не выполнено, либо выполнено не в срок и с грубыми ошибками. При процедуре защиты студент некорректно отвечает на большинство дополнительных вопросов, в том числе и на вопросы для самоконтроля. Не может объяснить этапы выполнения задания, характеристики и свойства полученного результата, причины и взаимосвязи между ними, исходными данными и своими действиями. Неспособен доработать полученные результаты в соответствии с незначительными изменениями в задании.

Шкала оценивания итогового тестирования:

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач				
ИУК-1.1. Анализирует задачу, выделяя ее базовые составляющие ИУК-1.2. Осуществляет поиск, критически оценивает, обобщает, систематизирует и ранжирует информацию, требуемую для решения поставленной задачи ИУК-1.3. Рассматривает и предлагает рациональные варианты решения поставленной задачи, используя системный подход, критически оценивает их достоинства и недостатки	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.
УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений				
ИУК-2.1. Формулирует совокупность задач в рамках поставленной цели проекта, решение	Обучающийся демонстрирует полное отсутствие или недостаточное	Обучающийся демонстрирует неполное соответствие следующих знаний,	Обучающийся демонстрирует частичное соответствие следующих знаний,	Обучающийся демонстрирует полное соответствие следующих знаний,

которых обеспечивает ее достижение ИУК-2.2. Определяет связи между поставленными задачами, основными компонентами проекта и ожидаемыми результатами его реализации ИУК-2.3. Выбирает оптимальные способы планирования, распределения зон ответственности, решения задач, анализа результатов с учетом действующих правовых норм, имеющихся условий, ресурсов и ограничений, возможностей использования	соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины	указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения				
ИОПК-7.1. Знает основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует

ИОПК-7.2. Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ИОПК-7.3. Владеет языком программирования, методами отладки и тестирования работоспособности программы.	компетенций дисциплины	ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	незначительные ошибки, неточности, затруднения при аналитических операциях.	приобретенными знаниями.

7.3 Оценочные средства

7.3.1 Текущий контроль

Оценочные средства текущего контроля успеваемости включают опросы, контрольные вопросы и тестирование (итоговое) для контроля освоения обучающимися разделов дисциплины.

Типовой экзаменационный билет

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

по дисциплине

«Прикладное программирование»

направление подготовки 09.03.03 Прикладная информатика

ВОПРОСЫ:

1. Особенности языка Java: интерпретируемость, независимость от платформы, мобильность. Примеры.

2. События в Java. Интерфейсы блоков прослушивания событий. Способы реализации блока прослушивания. Примеры.
3. Условный оператор *if* . Примеры.

Утверждено: _____ / _____ / «__» _____ 20__г.

Пример вопросов для экзамена

1. Особенности языка Java: интерпретируемость, независимость от платформы, мобильность. Примеры.
2. Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах). Виртуальная машина Java. JIT-компиляция. Примеры.
3. Управляющие последовательности. Символы Unicode. Специальные символы. Примеры.
4. Идентификаторы. Переменные и типы. Примитивные и ссылочные типы. Примеры.
5. Работа со ссылочными переменными. Сборка мусора в языке Java. Примеры.
6. Проекты. Пакеты. Уровни видимости классов. Импорт классов. Примеры.
7. Структура проекта NetBeans. Примеры.
8. Документирование исходного кода в Java. Примеры.
9. Отладка приложений в среде NetBeans. Примеры.
10. Встроенные типы данных. Способы задания литералов различных типов.
11. Булевый (логический) тип. Основные операторы для работы с логическими величинами. Примеры.
12. Целые типы, переменные, константы. Основные операторы для работы с целочисленными величинами. Примеры.
13. Вещественные типы и класс *Math.* Примеры.
14. Оператор присваивания. Порядок действий (приоритет операторов). Примеры.
15. Приведение типов. Неявное приведение типов. Явное приведение, автоматическое расширение типов и instanceof. Строковое приведение. Примеры.
16. Правила явного и автоматического преобразования типа при работе с числовыми величинами. Примеры.
17. Метки в языке Java. Примеры.
18. Управляющие инструкции. Операторы и блоки. Составной оператор. Примеры.
19. Условный оператор *if* . Примеры.
20. Оператор выбора *switch.* Примеры.
21. Условное выражение *...?... :* ... Примеры.
22. Операторы инкремента ++ и декремента -- Примеры.
23. Оператор цикла *for*. Примеры.
24. Цикл с предусловием. Оператор цикла *while*. Примеры.
25. Цикл с постусловием. Оператор цикла *do...while*. Примеры.
26. Операторы прерывания *continue*, *break*, *return*, *System.exit*. Примеры.
27. Арифметические операторы. Примеры.
28. Логические операторы. Операторы сравнения. Побитовые операторы. Примеры.
29. Оболочечные классы. Упаковка (*boxing*) и распаковка (*unboxing*). Примеры.
30. Массивы. Многомерные массивы. Примеры.
31. Присваивание и сравнение массивов. Примеры.
32. Коллекции, списки, итераторы. Примеры.

33. Перебор в цикле элементов коллекций. Оператор цикла *for-each*. Примеры.
34. Классы языка программирования Java. Методы и поля. Примеры.
35. Конструкторы. Конструктор по умолчанию. Зарезервированные слова *super* и *this*. Блоки инициализации. Примеры.
36. Конструкторы в наследуемых классах. Порядок вызова конструкторов. Примеры.
37. Удаление неиспользуемых объектов и метод *finalize*. Проблема деструкторов для сложно устроенных объектов. Примеры.
38. Модификаторы уровня доступа default, public, protected, private. Примеры.
39. Наследование. Суперклассы и подклассы. Переопределение методов. Примеры.
40. Наследование и правила видимости. Зарезервированное слово *super*. Примеры.
41. Проблемы множественного наследования классов. Интерфейсы. Примеры.
42. Отличия интерфейсов от классов. Проблемы наследования интерфейсов. Примеры.
43. Композиция как альтернатива множественному наследованию. Примеры.
44. Интерфейсы как средство реализации множественного наследования. Примеры.
45. Правила совместимости ссылочных типов как основа использования полиморфного кода. Приведение и проверка типов. Примеры.
46. Методы. Значения параметров. Применение методов для ограничения доступа. Примеры.
47. Сигнатура метода. Перегрузка методов. Примеры.
48. Статическое и динамическое связывание методов. Полиморфизм. Примеры.
49. Функции. Модификаторы. Передача примитивных типов в функции. Примеры.
50. Локальные и глобальные переменные. Модификаторы доступа и правила видимости. Ссылка *this*. Примеры.
51. Передача ссылочных типов в функции. Проблема изменения ссылки внутри подпрограммы. Примеры.
52. Статические (static) вложенные классы и интерфейсы. Примеры.
53. Внутренние (inner) классы. Примеры.
54. Локальные (local) классы. Примеры.
55. Анонимные (anonimous) классы и обработчики событий. Примеры.
56. Анонимные (anonimous) классы и слушатели событий (listeners). Примеры.
57. Статические члены класса. Примеры.
58. Абстрактные классы и методы. Примеры.
59. Исключительные ситуации. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Контролируемые и неконтролируемые исключения. Примеры.
60. Объявление типа исключительной ситуации и оператор *throw*. Примеры.
61. Объявление метода, который может возбуждать исключительную ситуацию. Зарезервированное слово *throws*. Примеры.
62. Создание новых типов исключений. Оператор throw. Условие throws. Примеры.
63. Операторы try, catch и finally. Примеры.
64. События в Java. Понятие события. Типы событий. Иерархия классов событий. Примеры.
65. События в Java. Модель делегирования событий. Примеры.
66. События в Java. Интерфейсы блоков прослушивания событий. Способы реализации блока прослушивания. Примеры.
67. События в Java. Использование внутренних классов, классов-адаптеров для упрощения обработки событий. Примеры.
68. Типы-перечисления (enum). Примеры.
69. Работа с датами и временем. Класс *GregorianCalendar*. Примеры.
70. Работа со строками в Java. Строки как объекты. Классы *String*, *StringBuffer* и *StringBuilder*. Примеры.

71. Операции со строками. Сравнение строк. Поиск и извлечение строк. Посимвольная обработка строк. Пословное реверсирование строки. Преобразование всех символов строки к верхнему или нижнему регистру. Примеры.

72. Классы для работы с текстом. Примеры.

73. Специальные символы регулярных выражений. Примеры.

74. Классы Java Matcher и Pattern, используемые для работы с регулярными выражениями. Примеры.

75. Нахождение соответствий в тексте с помощью регулярных выражений. Нахождение всех вхождений по образцу. Замена текста, отвечающего соответствию. Примеры.

76. Ввод/вывод в Java. Основные группы классов и интерфейсов пакета java.io. Примеры.

77. Стандартные потоки ввода-вывода. Организация ввода и вывода данных. Класс Scanner. Примеры.

78. Ввод/вывод в Java. Фильтрованные потоки. Примеры.

79. Ввод/вывод в Java. Буферизированные потоки. Примеры.

80. Ввод/вывод в Java. Канальные потоки. Примеры.

Типовые практические задания

Напишите на языке Java программы расчета по двум формулам с вводом исходных данных и выводом результатов. Исходные данные представляют последовательность из пяти чисел. В программе использовать все виды циклов. Одну программу написать как консольное приложение, другую – с использованием оконного интерфейса. Результаты расчета по двум формулам должны совпасть (показать почему).

Вариант 1

$$z_1 = 2\sin^2(3\pi - 2\alpha)\cos^2(5\pi + 2\alpha)$$

$$z_2 = \frac{1}{4} - \frac{1}{4}\sin\left(\frac{5}{2}\pi - 8\alpha\right)$$

Вариант 2

$$z_1 = \cos\alpha + \sin\alpha + \cos 3\alpha + \sin 3\alpha$$

$$z_2 = 2\sqrt{2}\cos\alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right)$$

Вариант 3

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos\alpha + 1 - 2\sin^2 2\alpha}$$

$$z_2 = 2\sin\alpha$$