Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 05.06.2024 16:31:45
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
«Информационные технологии»
_____ / Д.Г.Демидов /
«15» февраля 2024г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»

Направление подготовки
09.03.03 Прикладная информатика

Профиль
«Разработка и интеграция бизнес-приложений»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2024 г.

Разработчик рабочей программы дисциплины «Основы программирования» профиля «Разработка и интеграция бизнес-приложений» направления подготовки 09.03.03 «Прикладная информатика»:

26.04.2024

старший преподаватель /Е.А. Харченко/

Согласовано:

Заведующий кафедрой «Инфокогнитивные технологии»,
доцент, к.т.н. /Е.А.Пухова/

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость (по формам обучения) 5
3.2 Тематический план изучения дисциплины (по формам обучения) 6
3.3 Содержание дисциплины 7
3.4 Тематика семинарских/практических и лабораторных занятий 8
3.5 Тематика курсовых проектов (курсовых работ) 12
4 Учебно-методическое и информационное обеспечение 12
4.1 Нормативные документы и ГОСТы 12
4.2 Основная литература 12
4.3 Дополнительная литература 13
4.4 Электронные образовательные ресурсы 13
4.5 Лицензионное и свободно распространяемое программное обеспечение 13
4.6 Современные профессиональные базы данных и информационные справочные системы 13
5 Материально-техническое обеспечение 13
6 Методические рекомендации 13
6.1 Методические рекомендации для преподавателя по организации обучения 13
6.2 Методические указания для обучающихся по освоению дисциплины 14
7 Фонд оценочных средств 14
7.1 Методы контроля и оценивания результатов обучения 14
7.2 Шкала и критерии оценивания результатов обучения 14
7.3 Оценочные средства 15

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цели дисциплины «Основы программирования»: ознакомление студентов с методами построения программ и программирования в малом; ознакомление студентов с основами современных языков объектно-ориентированного программирования; формирование у студентов практических умений применения фундаментальных знаний в области технологии и практики современного программирования; формирование у студентов профессиональных навыков применения этих знаний.

Задачей дисциплины является обучение студентов методам построения программ и программирования в малом, выработка у студентов навыков владения современными языками объектно-ориентированного программирования, освоение студентами фундаментальных знаний в области технологии и практики современного программирования.

В результате изучения дисциплины студенты должны знать: понятия алгоритма и алгоритмического языка; базисные методы обработки информации; способы построения программ; основные концепции объектно-ориентированного программирования; базовые конструкции языка Java.

В результате изучения дисциплины студенты должны уметь: реализовывать простейшие алгоритмы обработки информации на языке Java; использовать различные типы данных; применять основные концепции объектно-ориентированного программирования.

В результате изучения дисциплины студенты должны владеть: навыками по созданию и модификации небольших программ; навыками разработки и отладки программ на алгоритмическом процедурном языке программирования высокого уровня; навыками проверки корректности и работоспособности программ.

Обучение по дисциплине «Основы программирования» направлено на формирование у обучающихся следующих общеобразовательных и профессиональных компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
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.
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 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.
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 «Основы программирования» относится к части, формируемой участниками образовательных отношений, блока Б1 «Дисциплины (модули)».

Дисциплина базируется на знаниях и навыках, полученных студентами в средней школе. В частности, предполагается, что студенты знакомы с основными понятиями информатики и каким-либо языком программирования высокого уровня, а также что студенты имеют общее представление о структуре современного компьютера и операционных системах.

Приобретенные в ходе изучения дисциплины «Основы программирования» знания, навыки и умения в большей мере требуются для освоения дисциплин: «Прикладное программирование», «Основы тестирования», «Алгоритмы и структуры данных», «Шаблоны проектирования приложений», «Разработка мобильных приложений», «Проектная деятельность».

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины «Основы программирования» составляет 5 зачетных единиц (184 часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры
			1
1	**Аудиторные занятия**	**48**	48
	В том числе:		
1.1	Лекции	8	8
1.2	Семинарские/практические занятия	0	0
1.3	Лабораторные занятия	40	40
2	**Самостоятельная работа**	**136**	136
3	**Промежуточная аттестация**		
	Курсовой проект	–	**КП**

	Экзамен	–	**Экзамен**
	Итого:	**180**	180

3.1.2 Очно-заочная форма обучения

Не предусмотрена программой.

3.1.3 Заочная форма обучения

Не предусмотрена программой.

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	Раздел 1. Основы языка программирования Java						
1.1	Тема 1. Java как язык программирования общего назначения	8	0	0	2	0	6
1.2	Тема 2. Переменные, примитивные и ссылочные типы данных	8	0	0	2	0	6
1.3	Тема 3. Статические методы как функции	8	0	0	2	0	6
1.4	Тема 4. Условные операторы и циклы	8	0	0	2	0	6
1.5	Тема 5. Задачи на последовательности	8	0	0	2	0	6
1.6	Тема 6. Одномерные и двухмерные массивы	10	0	0	4	0	6
1.7	Тема 7. Строки и файлы	8	0	0	2	0	6
1.8	Тема 8. Рекурсия	8	0	0	2	0	6
1.9	Тема 9. Класс и экземпляры класса	13	1	0	2	0	10
1.10	Тема 10. Модификаторы доступа	13	1	0	2	0	10
1.11	Тема 11. Переменные и методы экземпляра класса	13	1	0	2	0	10
1.12	Тема 12. Переменные и методы класса	13	1	0	2	0	10
1.13	Тема 13. Инкапсуляция, наследование, полиморфизм	13	1	0	2	0	10
1.14	Тема 14. Перегрузка и переопределение метода	13	1	0	2	0	10

1.15	Тема 15. Абстрактные классы и интерфейсы	13	1	0	2	0	10
1.16	Тема 16. Разработка классов	25	1	0	6	0	18
1.17	Тема 17. Обзор пройденного материала	2	0	0	2	0	0
Итого		**184**	**8**	**0**	**40**	**0**	**136**

3.2.2 Очно-заочная форма обучения
Не предусмотрена программой.

3.2.3 Заочная форма обучения
Не предусмотрена программой.

3.3 Содержание дисциплины

Раздел 1. Основы языка программирования Java

Тема 1. Java как язык программирования общего назначения

Объектно-ориентированный язык программирования Java: кроссплатформенность, строгая типизация, мультифункциональность. Структура программы на Java, ключевое слово class. Точка входа в программу, метод main. Первая программа: «Hello, World!». Компиляция и запуск программы (через командную строку, через среду разработки). Комментарии.

Тема 2. Переменные, примитивные и ссылочные типы данных

Понятие переменной, объявление и инициализация переменной. Оператор присваивания =. Примитивные типы данных: byte, short, int, long, float, double, boolean, char. Арифметические операторы над данными примитивных типов и их приоритеты: +, -, *, /, %. Совмещенные операторы: +=, -=, *=, /=, %=. Приведение типов. Инкременты (постфиксные и префиксные): ++, --. Работа со стандартными потоками вывода и ввода. Форматированный вывод. Понятия класса и объекта, или экземпляра, класса (на примере строк и массивов). Ссылочные типы данных. Оператор проверки типа переменной instanceof. Примеры программ.

Тема 3. Статические методы как функции

Спецификация метода. Не зависящие и зависящие от аргументов методы. Не возвращающие и возвращающие значения методы. Ключевые слова void и return. Вложенные вызовы методов. Математическая библиотека Math, ключевое слово import. Примеры программ.

Тема 4. Условные операторы и циклы

Логические выражения. Логические операторы: &, &&, |, ||, !. Операторы сравнения: >, <, >=, <=, ==, !=. Условные операторы: if, if-else, тернарный условный оператор, switch. Циклы: for, while, do-while. Ключевые слова break и continue. Вложенные условные операторы и циклы. Примеры программ.

Тема 5. Задачи на последовательности

Бесконечные циклы. Обработка исключительных ситуаций, блоки try и catch. Генерация исключений, ключевые слова throw и throws. Пример: поиск количества максимальных элементов в сколь угодно длинной последовательности вводимых с клавиатуры целых чисел.

Тема 6. Одномерные и двухмерные массивы

Понятие массива. Способы создания и заполнения массива. Итератор (for-each). Двухмерный массив как одномерный массив одномерных массивов. Примеры программ.

Тема 7. Строки и файлы

Массив символов. Строка (объект класса String), методы работы со строками. Работа с файлом как с символьным потоком: чтение из файла (через объект класса FileReader) и запись в файл (через объект класса FileWriter). Примеры программ.

Тема 8. Рекурсия

Рекурсия и итерация. Особенности рекурсивных программ. Примеры программ.

Тема 9. Класс и экземпляры класса: конструкторы класса

Структура класса. Конструкторы класса. Создание объекта класса. Разработка учебного проекта Birds.

Тема 10. Модификаторы доступа

Модификаторы доступа: по умолчанию, public, private, protected. Разработка учебного проекта Birds.

Тема 11. Переменные и методы экземпляра класса

Переменные (атрибуты) и методы экземпляра класса. Ключевое слово this. Геттеры, сеттеры. Разработка учебного проекта Birds.

Тема 12. Переменные и методы класса

Переменные и методы класса. Ключевое слово static. Разработка учебного проекта Birds.

Тема 13. Инкапсуляция, наследование, полиморфизм

Свойства объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Ключевое слово super. Класс Object. Ключевое слово final. Разработка учебного проекта Birds.

Тема 14. Перегрузка и переопределение метода

Перегрузка метода. Переопределение метода. Разработка учебного проекта Birds.

Тема 15. Абстрактные классы и интерфейсы

Абстрактный класс, ключевое слово abstract. Интерфейс, ключевое слово interface. Разработка учебного проекта Birds.

Тема 16. Разработка классов

Разбор домашних заданий. Модификация проекта Bird, использование собственных классов и встроенных классов графической библиотеки Swing. Непрерывные и ссылочные реализации контейнеров. Разбор примеров из списка литературы.

Тема 17. Обзор пройденного материала

Парадигма объектно-ориентированного программирования.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

Не предусмотрены программой.

3.4.2 Лабораторные занятия

Тема работы 1. Решение практических задач на примитивные типы данных и методы как функции

Цель работы: изучение теоретических основ использования в программах примитивных типов данных и методов (функций); выработка практических навыков написания программ с использованием примитивных типов данных и методов (функций). Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 2. Решение практических задач на условные операторы и циклы

Цель работы: изучение теоретических основ использования в программах условных операторов и циклических конструкций; выработка практических навыков написания программ с использованием условных операторов и циклических конструкций. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 3. Решение практических задач на последовательности

Цель работы: изучение теоретических основ обработки числовых последовательностей в программах; выработка практических навыков написания программ, обрабатывающих числовые последовательности. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 4. Решение практических задач на одномерные и двухмерные массивы

Цель работы: изучение теоретических основ использования в программах одномерных и двухмерных массивов; выработка практических навыков написания программ с использованием одномерных и двухмерных массивов. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 5. Решение практических задач на одномерные и двухмерные массивы

Цель работы: изучение теоретических основ использования в программах одномерных и двухмерных массивов; выработка практических навыков написания программ с использованием одномерных и двухмерных массивов. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 6. Решение практических задач на массивы символов и строки

Цель работы: изучение теоретических основ использования в программах массивов символов и строк; выработка практических навыков написания программ с использованием массивов символов и строк. Содержание работы: решение разноуровневых задач по теме

работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 7. Решение практических задач на файлы

Цель работы: изучение теоретических основ обработки данных из файлов в программах; выработка практических навыков написания программ с обработкой данных из файлов. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 8. Решение практических задач на рекурсию

Цель работы: изучение теоретических основ построения рекурсивных программ; выработка практических навыков написания рекурсивных программ. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 9. Решение практических задач на разработку простых объектно-ориентированных программ

Цель работы: изучение теоретических основ построения и разработки программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 10. Решение практических задач на разработку простых объектно-ориентированных программ

Цель работы: изучение теоретических основ построения и разработки программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 11. Решение практических задач на разработку простых объектно-ориентированных программ

Цель работы: изучение теоретических основ построения и разработки программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита

студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 12. Решение практических задач на разработку простых объектно-ориентированных программ

Цель работы: изучение теоретических основ разработки программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 13. Решение практических задач на модификацию простых объектно-ориентированных программ

Цель работы: изучение теоретических основ разработки и модификации программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания и модификации программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 14. Решение практических задач на модификацию простых объектно-ориентированных программ

Цель работы: изучение теоретических основ разработки и модификации программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания и модификации программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 15. Решение практических задач на модификацию простых объектно-ориентированных программ

Цель работы: изучение теоретических основ разработки и модификации программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания и модификации программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы. Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

Тема работы 16. Решение практических задач на модификацию простых объектно-ориентированных программ

Цель работы: изучение теоретических основ разработки и модификации программ в парадигме объектно-ориентированного программирования; выработка практических навыков написания и модификации программ в парадигме объектно-ориентированного программирования. Содержание работы: решение разноуровневых задач по теме работы.

Пример задания представлен в разделе 7.3. Результат работы студента: разработанные согласно заданию программы. Подготовка студента к работе, выполнение студентом работы и защита студентом работы направлены на формирование у студента компетенций УК-1, УК-2 и ОПК-7.

3.5 Тематика курсовых проектов (курсовых работ)

Вариативная тема курсового проекта: «Основы объектно-ориентированного программирования. Реализация игры-симулятора «Жизнь». Исходные тестовые данные выдаются по вариантам.

Цель работы: разработка отвечающего парадигме объектно-ориентированного программирования приложения, которое моделирует эволюционные изменения колонии живых организмов и может быть использовано как основа для решения различных прикладных задач (например, в области теории информации или компьютерного дизайна).

Уточненная тема работы вырабатывается студентом в результате изучения им предложенной предметной области, тема должна отражать вариант практического применения реализованного в работе математического аппарата.

Примеры простых начальных состояний колоний живых организмов: «Блок», «Улей», «Колодец», «Каравай», «Баржа», «Лодка», «Корабль», «Навигационные огни», «Тумблер», «Глайдер». Примеры сложных начальных состояний колоний живых организмов: «Кактус», «Спираль», «Лабиринт», «Ружье», а также комбинации простых.

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.

4.2 Основная литература

1. Пономарчук, Ю.В. Программирование на языке Java: учебное пособие / Ю.В. Пономарчук, И.В. Кузнецов. – Хабаровск: ДВГУПС, 2021. – 103 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/259451 (дата обращения: 28.09.2023). – Режим доступа: для авториз. пользователей.

2. Строганкова, Н.В. Шаблоны программных платформ языка Java: учебное пособие / Н.В. Строганкова, К.В. Касьяненко, А.В. Хозяинов; составители практикума входят:

использование. – Москва: РТУ МИРЭА, 2021. – 83 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/182466 (дата обращения: 28.09.2023). – Режим доступа: для авториз. пользователей.

4.3 Дополнительная литература

1. Хорстманн К. С. Java. Библиотека профессионала. Том 1. Основы. – СПб.: ООО "Диалектика", 2020. – 864 с.
2. Шилдт Г. Java. Руководство для начинающих. – СПб.: ООО "Диалектика", 2019. – 816 с.
3. Эккель Б. Философия Java. – СПб.: Питер, 2018. – 1168 с.
4. Шилдт Г. Java. Полное руководство. – М.: ООО "И.Д. Вильямс", 2022. – 1344 с.
5. Лафоре Р. Структуры данных и алгоритмы в Java. – СПб.: Питер, 2018. – 704 с.

4.4 Электронные образовательные ресурсы

1. Java. Базовый курс. – URL: https://stepik.org/courses/187
2. Основы информатики и программирования. – URL: https://intuit.ru/studies/courses/105/105/info

Соответствующий настоящей рабочей программе дисциплины электронный образовательный ресурс находится в разработке.

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Браузер, Atom (как пример): https://browser.ru
2. Комплект программ JDK (Java Development Kit): https://www.oracle.com/java/technologies/downloads
3. Интегрированная среда разработки, Intellij IDEA Community Edition (как пример): https://www.jetbrains.com/idea/download

4.6 Современные профессиональные базы данных и информационные справочные системы

1. Java Documentation. – URL: https://docs.oracle.com/en/java
2. Intellij IDEA Documentation. – URL: https://www.jetbrains.com/help/idea

5 Материально-техническое обеспечение

Требования к оборудованию в аудиториях для проведения лекций: доска, проектор. Требования к аудиториям для проведения лабораторных занятий: компьютерный класс с доступом в интернет. Необходимое для проведения лекций и лабораторных занятий программное обеспечение приведено в разделе 4.5.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

Методические рекомендации по организации лабораторного занятия: 1) обзор теоретического минимума для выполнения лабораторной работы; 2) разбор алгоритмов и примеров по теме работы; 3) консультация по тематике работы; 4) предварительные проверка работы и доведение до студента ошибок или недочетов работы; 5) прием работы с вопросами по решению и заданиями на модификацию решения.

Выполнение студентом курсового проекта должно сопровождаться консультационной поддержкой преподавателя. Поэтапный контроль качества выполнения задания проекта желателен. Рекомендуемые сроки доведения задания до студентов – за полтора месяца до конца семестра.

6.2 Методические указания для обучающихся по освоению дисциплины

Методические указания по освоению дисциплины: 1) предварительное изучение теоретического материла лабораторных работ с обращением к источникам из списка литературы; 2) последовательное и своевременное выполнение и защита лабораторных работ; 3) выполнение и документирование задания курсового проекта.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Виды текущего контроля успеваемости студентов: 1) проверка и прием лабораторных работ; 2) проверка и прием курсового проекта. Вид промежуточной аттестации – экзамен.

В рамках каждой лабораторной работы студенту предлагается для решения несколько практических разноуровневых задач на определенную тему. На выполнение одной лабораторной работы отводится одна неделя. Все работы подлежат защите.

В рамках курсового проекта студенту предлагается разработать простое консольно-оконное приложение на языке программирования Java, а также составить пояснительную записку к выработанному решению. На выполнение задания отводится полтора месяца. Работа подлежит защите и должна быть предоставлена для предварительной проверки не позднее, чем за сутки до дня защиты работы.

В рамках экзамена студенту предлагается выполнить за ограниченное время (60 минут) два практических задания по билету, а также ответить на уточняющие и дополнительные вопросы по решениям и по тематике курса. При подготовке к ответу и непосредственно при ответе студенту разрешается использовать только среду разработки приложений на языке Java.

7.2 Шкала и критерии оценивания результатов обучения

Для оценки результатов обучения используется балльно-рейтинговая система. Экзаменационная оценка определяется суммой баллов, набранных студентом за выполнение и защиту лабораторных работ в течение семестра и сдачу экзамена. Вклад лабораторных работ в итоговую оценку – до 81 баллов, экзаменационных заданий – до 19 баллов. Правила перевода совокупных баллов в экзаменационную оценку следующие: сумма баллов от 0 до 40 – оценка «неудовлетворительно»; сумма баллов от 41 до 60 баллов – оценка «удовлетворительно»; сумма баллов от 61 до 80 баллов – оценка «хорошо»; сумма баллов от 81 до 100 баллов – оценка «отлично».

Первично результаты выполнения и защиты каждой лабораторной работы оцениваются в баллах по следующей шкале: 3 балла – компетенции считаются освоенными на продвинутом уровне; 2 балла – компетенции считаются освоенными на базовом уровне; 1 балл – компетенции считаются не освоенными. Полученные баллы умножаются на коэффициент сложности: для работ 1, 2, 4, 6-8, 16 коэффициент сложности равен 1; для работ 5, 10-15 коэффициент сложности равен 2; для работ 3, 9 коэффициент сложности равен 3. Работы 11, 14 и 16 выполняются исключительно на занятиях.

Результаты выполнения экзаменационных заданий с учетом ответов на уточняющие и дополнительные вопросы по курсу оцениваются по шкале: от 0 до 8 баллов – за первое задание; от 0 до 11 – за второе задание. Детализация баллов за первое задание: от 7 до 8 баллов – решение и ответ не содержат ошибок или существенных недочетов; от 5 до 6 баллов – решение и ответ не содержат ошибок, но имеют недочеты; от 3 до 4 баллов –

решение или ответ содержат ошибки; от 0 до 2 баллов – задание не выполнено или не даны ответы на вопросы по решению. Детализация баллов за второе задание: от 10 до 11 баллов – решение и ответ не содержат ошибок или существенных недочетов; от 7 до 9 баллов – решение и ответ не содержат ошибок, но имеют недочеты; от 4 до 6 баллов – решение или ответ содержат ошибки; от 0 до 3 баллов – задание не выполнено или не даны ответы на вопросы по решению.

При повторной промежуточной аттестации необходимым условием получения студентом положительной экзаменационной оценки является получение им не менее 11 баллов за выполнение экзаменационных заданий (с учетом ответов на вопросы).

Оценка за курсовой проект рассчитывается как среднее оценок за приложение и пояснительную записку с учетом ответов на вопросы по решению и отчету. Критерии совокупной оценки и каждой из ее составляющих: «отлично» – работа выполнена в полном объеме без ошибок или существенных недочетов; «хорошо» – работа выполнена в полном объеме, но содержит недочеты; «удовлетворительно» – работа выполнена в полном объеме, но содержит ошибки; «неудовлетворительно» – работа не выполнена или выполнена не в полном объеме. Необходимым условием получения положительной оценки за курсовой проект является выполнение обеих составляющих работы.

7.3 Оценочные средства

7.3.1 Примеры заданий лабораторного практикума

Пример задания из лабораторной работы 1.

Реализуйте метод, вычисляющий и возвращающий расстояние между двумя точками на плоскости.

Пример задания из лабораторной работы 2.

Роботизированная машина может передвигаться вперед и назад по прямой дороге, размеченной цифрами от 0 до 99. Реализуйте метод, который может «управлять» роботом, получая на вход команду «вперед» или «назад», число ячеек для перемещения робота и его текущую позицию. Робот не может съехать с дороги, поэтому если он достигает края дороги, раньше, чем пройдет заданное число ячеек, он остается в крайнем положении. Метод должен возвращать текущее положение робота.

Пример задания из лабораторной работы 3.

Напишите программу, вводящую последовательность из нулей и единиц, печатающую число групп из единиц, разделенных нулями (без использования массива или строки).

Пример задания из блока лабораторных работ 4, 5.

Напишите программу, которая создает двухмерный массив из m строк и n столбцов, заполняет его случайными числами от 0 до 9 и преобразует, меняя местами строки, содержащие первый минимальный и первый максимальный элементы матрицы.

Пример задания из лабораторной работы 6.

Реализуйте метод, определяющий первое вхождение заданной подстроки в заданную строку. Метод должен возвращать индекс первого символа первого вхождения подстроки в строку или -1, если подстрока в строке не содержится.

Пример задания из лабораторной работы 7.

Текстовый файл содержит строки вида: «фамилия имя отчество:дата рождения:номер телефона». Напишите программу, которая выводит имена и отчества людей старше 20 лет (когда бы программа ни запускалась). Файл может содержать нежелательные или

избыточные пробельные символы, неполные сведения и пустые строки. Строки неправильного формата при обработке следует опускать.

Пример задания из лабораторной работы 8.

Реализуйте метод, вычисляющий наибольший общий делитель двух натуральных чисел: а) итерационно; б) рекурсивно.

Пример задания из блока лабораторных работ 9-12.

Реализуйте класс Matrix, описывающий матрицу и позволяющий выполнять следующие операции над матрицами (объектами класса): 1) сложение двух матриц (как метод класса и как метод экземпляра класса); 2) умножение матрицы на число (как метод класса и как метод экземпляра класса); 3) умножение матрицы на матрицу (как метод класса и как метод экземпляра класса).

Пример задания из блока лабораторных работ 13-16.

Модифицируйте проект Birds так, чтобы все изображаемые птицы очерчивались минимальным прямоугольником. Рамка должна проходить через центры крайних птиц.

7.3.2 Вопросы и задания для самопроверки

1. Какой размер имеет переменная типа `int`?

а) байт
б) 2 байта
в) 4 байта
г) 8 байт

2. Какие из перечисленных типов данных относятся к примитивным?

а) `char[]`
б) `Integer`
в) `double`
г) `String`

3. Имеется фрагмент кода программы:

```
int a = 3;
int b = 5;
b += a;
a = b - a;
b = b - a;
System.out.println(b);
```

Что выведет программа?

а) 5
б) 2
в) 8
г) 3

4. Имеется фрагмент кода программы:

```
int i = 3;
System.out.println(i++);
```

Что выведет программа?

а) 3
б) 4
в) 5
г) Сообщение об ошибке

5.Имеется фрагмент кода программы:

```
System.out.println(16/5%2);
```

Что выведет программа?

а) 0.2

б) 1

в) 5

г) 3

6. Имеется фрагмент кода программы:

```
int x = 15;
if(x < 10){
	System.out.println(1);
}
else if(x > 10){
	System.out.println(2);
}
else{
	System.out.println(3);
}
```

Что выведет программа?

а) Ничего

б) 2

в) 3

г) 2, 3

7. Имеется фрагмент кода программы:

```
int i = 0;
while(i < 5){
	++i;
}
System.out.println(i);
```

Что выведет программа?

а) Ничего, цикл не завершится

б) 5

в) 4

г) Сообщение об ошибке

8. Имеется фрагмент кода программы:

```
int[] a = new int[]{8, 2, 4, 5};
int[] b = a;
a[2] = 10;
b[2] = 0;
System.out.println(a[2]);
```

Что выведет программа?

а) 10

б) 4

в) 0

г) 2

9. Имеется фрагмент кода программы:

```
int[][] a = new int[5][5];
for(int i = 0; i < 5; i++)
	a[i][i] = 1;
System.out.println(a[4][0]);
```

Что выведет программа?

а) 0

б) Сообщение об ошибке

в) 1

г) null

10. Имеется фрагмент кода программы:

```
static String f(int x){
      x = x + 1;
      return "" + x;
}
```

Какие из перечисленных способов вызова метода корректны?

а) Math.sqrt(f(3))

б) f(f(3))

в). System.out.println(f(3))

г). f(3) + f(3)

11. Имеется фрагмент кода программы:

```
static int f(int k){
      return k > 0 ? k + f(k - 1) : 0;
}
```

Какое значение вернет метод, если его вызвать как f(5)?

а) 10

б) 0

в) 9

г) 15

12. Имеется фрагмент кода программы:

```
System.out.println('j' > 'a' || 'v' < 'a' && 'a' == 'a');
```

Что выведет программа?

а). 0

б) true

в) Сообщение об ошибке

г) false

13. Имеется фрагмент кода программы:

```
String s = "110110011101";
System.out.println(s.split("0").length);
```

Что выведет программа?

а) 4

б) 5

в) 3

г) 11

14. Имеется фрагмент кода программы:

```
FileReader fr = new FileReader("file.txt");
Scanner in = new Scanner(fr);
while(fr.hasNextLine()){
      String[] ar = in.nextLine().replaceAll(",|\\.|-", "").split();
      for(int i = 0; i < ar.length; i++)
            System.out.println(ar[i].length());
}
fr.close();
```

Текстовый файл, обрабатываемый программой, содержит текст:

Конструктор - это метод, имеющий имя, совпадающее с именем класса. Две основные задачи конструктора заключаются в выделении памяти под вновь создаваемый объект и его инициализация.

Что выведет программа?

а) Количество предложений.

б) Количество букв в каждом слове.

в) Количество слов в каждом предложении.

г) Сообщение об ошибке.

15. Какие из перечисленных ключевых слов не относятся к модификаторам доступа?

а) `static`

б) `final`

в) `protected`

г) `public`

16. Какие из приведенных утверждений верны?

а) Абстракция – принцип объектно-ориентированного программирования (ООП), согласно которому объекты разного рода объединяются общим понятием (концепцией), а затем сгруппированные сущности рассматриваются как элементы единой категории.

б) Инкапсуляция – принцип ООП, согласно которому несущественная с точки зрения интерфейса объекта информация прячется внутри него.

в) Наследование – принцип ООП, согласно которому экземпляры класса получают доступ к данным и методам классов-предков без их повторного определения.

г) Полиморфизм – принцип ООП, согласно которому для различных действий можно использовать один и тот же интерфейс.

17. Имеются фрагменты кода программы:

```
class A{
	private int a;

	public A(){
		System.out.println(a);
	}
}

class B{
	private int b;

	public B(int b){
		this.b = b;
		System.out.println(b);
	}
}

class Main{
	public static void main(String[] args){
		A a = new A();
		B b = new B(2);
		B b = a;
	}
}
```

Что выведет программа?

а) Сообщение об ошибке

б) 0

в) 2

г) 0, 2

18. Имеются фрагменты кода программы:

```
class A{
	private int a;

	public A(){
		System.out.println(a);
	}

	public void f(){
		System.out.println(10);
	}
}

class B extends A{
	private int b;

	public B(int b){
		this.b = b;
		System.out.println(b);
	}
}

class Main{
	public static void main(String[] args){
		B b = new B(2);
		b.f();
	}
}
```

Что выведет программа?

а) 0, 2, 10

б) 2, 10

в) Сообщение об ошибке

г) 0, 10

19. Имеются фрагменты кода программы:

```
class A{
	private int a;

	protected void f(){
		System.out.println(10);
	}
}

class B extends A{
	private int b;

	public B(int b){
		this.b = b;
		System.out.println(b);
	}

	@Override
	public void f(){
		System.out.println(20);
	}
}

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

```
            B b = new B(2);
            b.f();
        }
}
```

Что выведет программа?

а) 2, 10

б) 2, 10, 20

в) Сообщение об ошибке

г) 2, 20

20. Имеются фрагменты кода программы:

```
class A{
        public static int x = 0;

        public A(){
                x += 1;
        }

        public A f(int x){
                this.x += x;
                return this;
        }

        public static A f(A a, int x){
                a.x += x;
                return a;
        }
}

class Main{
        public static void main(String[] args){
                A a1 = new A();
                A a2 = new A();
                a2 = a1.f(1);
                a1 = a2.f(1);
                A.f(a1, 10);
                System.out.println(A.x);
        }
}
```

Что выведет программа?

а) 14

б) Сообщение об ошибке

в). 4

г). 1

7.3.3 Вопросы и задания для подготовки к защите лабораторных работ

Вопросы к работе 1:

1. Объясните назначение метода main.
2. Перечислите и опишите примитивные типы данных.
3. Дайте определение и приведите пример переменной.
4. Перечислите с указанием приоритета арифметические операции над данными примитивных типов.
5. Приведите примеры приведения типов данных.
6. Приведите примеры применения префиксного и постфиксного инкрементов.
7. Опишите способы взаимодействия со стандартными потоками ввода и вывода.
8. Приведите примеры форматированного вывода.
9. Приведите примеры переменных, имеющих ссылочные типы данных.

10. Дайте определение функции (статическому методу). Дайте определение и приведите пример спецификации метода.

11. Приведите пример и опишите принцип действия не зависящего от аргументов и не возвращающего значение метода.

12. Приведите пример и опишите принцип действия не зависящего от аргументов и возвращающего значение метода.

13. Приведите пример и опишите принцип действия зависящего от аргументов и не возвращающего значение метода.

14. Приведите пример и опишите принцип действия зависящего от аргументов и возвращающего значение метода.

15. Приведите пример вложенного вызова метода.

16. Приведите примеры использования констант и методов библиотеки Math.

17. Объясните структуру разработанных программ.

18. Объясните детали реализации разработанных программ.

19. Объясните порядок выполнения разработанных программ.

20. Расширьте функционал разработанных программ.

Вопросы к работе 2 (включают вопросы к работе 1):

1. Дайте определение логического выражения.

2. Перечислите операторы сравнения.

3. Перечислите логические операторы с указанием их приоритета.

4. Приведите примеры сложных (составных) логических выражений.

5. Перечислите и опишите принцип действия операторов ветвления.

6. Приведите пример использования оператора ветвления.

7. Перечислите и опишите принцип действия циклических конструкций.

8. Приведите пример использования цикла.

9. Приведите пример использования цикла с вложенным условным оператором.

10. Объясните структуру разработанных программ.

11. Объясните детали реализации разработанных программ.

12. Объясните порядок выполнения разработанных программ.

13. Расширьте функционал разработанных программ.

Вопросы к работе 3 (включают вопросы к работе 2):

1. Объясните принцип действия и приведите пример использования бесконечного цикла.

2. Приведите пример обработки бесконечной числовой последовательности.

3. Приведите пример способа организации бесконечного цикла, гарантирующего его корректное завершение.

4. Объясните механизм обработки и приведите пример использования исключительных ситуаций.

5. Приведите пример и объясните порядок выполнения программы, обрабатывающей бесконечную числовую последовательность.

6. Объясните структуру разработанных программ.

7. Объясните детали реализации разработанных программ.

8. Объясните порядок выполнения разработанных программ.

9. Расширьте функционал разработанных программ.

Вопросы к работам 4, 5 (включают вопросы к работе 3):

1. Дайте определение и приведите пример одномерного массива.

2. Приведите несколько способов заполнения одномерного массива.

3. Приведите несколько способов вывода элементов одномерного массива.

4. Дайте определение и приведите пример двухмерного массива.

5. Приведите несколько способов заполнения двухмерного массива.

6. Приведите несколько способов вывода элементов двухмерного массива.
7. Объясните структуру разработанных программ.
8. Объясните детали реализации разработанных программ.
9. Объясните порядок выполнения разработанных программ.
10. Расширьте функционал разработанных программ.

Вопросы к работе 6 (включают вопросы к работам 4, 5):
1. Объясните, как кодируются и сравниваются символы.
2. Дайте определение и приведите пример массива символов.
3. Приведите примеры строк (объектов класса String) и перечислите основные методы работы со строками.
4. Приведите способы преобразования массива символов к строке и строки к массиву символов.
5. Объясните структуру разработанных программ.
6. Объясните детали реализации разработанных программ.
7. Объясните порядок выполнения разработанных программ.
8. Расширьте функционал разработанных программ.

Вопросы к работе 7 (включают вопросы к работе 6):
1. Объясните порядок и приведите пример считывания символьных данных из файла.
2. Объясните порядок и приведите пример записи символьных данных в файл.
3. Объясните структуру разработанных программ.
4. Объясните детали реализации разработанных программ.
5. Объясните порядок выполнения разработанных программ.
6. Расширьте функционал разработанных программ.

Вопросы к работе 8 (включают вопросы к работе 7):
1. Дайте определение и приведите примеры рекурсии.
2. Приведите пример преобразования рекурсивного алгоритма к итерационному.
3. Объясните структуру разработанных программ.
4. Объясните детали реализации разработанных программ.
5. Объясните порядок выполнения разработанных программ.
6. Расширьте функционал разработанных программ.

Вопросы к работам 9-16 (включают вопросы к работе 8):
1. Опишите общую структуру класса.
2. Объясните назначение конструктора класса.
3. Приведите примеры создания объектов класса.
4. Перечислите модификаторы доступа.
5. Дайте определение и приведите пример переменной экземпляра класса.
6. Дайте определение и приведите пример метода экземпляра класса.
7. Дайте определение и приведите пример переменной класса.
8. Дайте определение и приведите пример метода класса.
9. Дайте определение и приведите пример перегрузки метода.
10. Дайте определение и объясните на примере принцип инкапсуляции.
11. Дайте определение и приведите пример использования механизма наследования.
12. Дайте определение и приведите пример полиморфизма.
13. Дайте определение и приведите пример переопределения метода.
14. Объясните назначение ключевого слова this.
15. Объясните назначение ключевого слова super.
16. Объясните назначение ключевого слова abstract.
17. Объясните назначение ключевого слова final.
18. Опишите возможности графической библиотеки Swing, приведите примеры.

19. Объясните структуру разработанных программ.
20. Объясните детали реализации разработанных программ.
21. Объясните порядок выполнения разработанных программ.
22. Расширьте функционал разработанных программ.

7.3.4 Вопросы к экзамену

1. Java как язык программирования общего назначения.
2. Переменные, примитивные и ссылочные типы данных.
3. Статические методы как функции.
4. Условные операторы и циклы.
5. Задачи на последовательности.
6. Одномерные и двухмерные массивы.
7. Строки и файлы.
8. Рекурсия.
9. Класс и экземпляры класса: конструкторы класса.
10. Модификаторы доступа.
11. Переменные и методы экземпляра класса.
12. Переменные и методы класса.
13. Инкапсуляция, наследование, полиморфизм.
14. Перегрузка и переопределение метода.
15. Абстрактные классы и интерфейсы.
16. Разработка классов.

7.3.5 Пример экзаменационного билета

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет «Информационные технологии», кафедра «Инфокогнитивные технологии»
Дисциплина «Основы программирования»
Направление 09.03.03 «Прикладная информатика», профиль «Разработка и интеграция бизнес-приложений»
Курс 1, семестр 1

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Напишите программу, вводящую сколь угодно длинную последовательность целых чисел и вычисляющую количество элементов, равных наибольшему из введенных.

   Замечание: общее количество элементов последовательности не запрашивается, последовательность вводится поэлементно, введенная часть последовательности не хранится, ответ выводится после каждого введенного числа.

2. Модифицируйте учебный проект «Birds» так, чтобы для заданной окружности (по введенным с клавиатуры радиусу и центру) вычислялось число попадающих в нее птиц.

   Замечание: недопустимы изменения исходного кода, не имеющие отношения к поставленной задаче.

Утверждено на заседании кафедры « » 2024 г., протокол № .

Зав. кафедрой _____/ /