Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 09.12.2025 12:51:11
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____ / Д.Г. Демидов /
«__» февраля 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»

Направление подготовки
09.03.03 Прикладная информатика

Профиль
«Разработка и интеграция бизнес-приложений»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2025 г.

Разработчик(и):



ст.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 5
3.2 Тематический план изучения дисциплины 6
3.3 Содержание дисциплины 7
3.4 Тематика семинарских/практических и лабораторных занятий 8
3.5 Тематика курсовых проектов (курсовых работ) 9
4 Учебно-методическое и информационное обеспечение 9
4.1 Нормативные документы и ГОСТы 9
4.2 Основная литература 10
4.3 Дополнительная литература 10
4.4 Электронные образовательные ресурсы 10
4.5 Лицензионное и свободно распространяемое программное обеспечение 10
4.6 Современные профессиональные базы данных и информационные справочные системы 10
5 Материально-техническое обеспечение 10
6 Методические рекомендации 11
6.1 Методические рекомендации для преподавателя по организации обучения 11
6.2 Методические указания для обучающихся по освоению дисциплины 11
7 Фонд оценочных средств 11
7.1 Методы контроля и оценивания результатов обучения 11
7.2 Шкала и критерии оценивания результатов обучения 11
7.3 Оценочные средства 12

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цели дисциплины «Основы программирования» – ознакомление студентов с методами построения программ и программирования в малом, современными языками объектно-ориентированного программирования, освоение фундаментальных знаний в области технологии и практики современного программирования и выработка практических навыков применения этих знаний.

Задачей дисциплины является обучение студентов методам построения программ и программирования в малом, выработка навыков владения современными языками объектно-ориентированного программирования, освоение 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 и основы 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 компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
УК1. 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.
УК2. 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.
ОПК7. 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)».

Базируется на знаниях, полученных студентами в средней школе. В частности, предполагается знакомство с основными понятиями информатики, каким-либо языком программирования высокого уровня, наличие общего представления о структуре современного компьютера и операционных системах.

Приобретенные в ходе изучения дисциплины «Основы программирования» знания в большей мере требуются для освоения таких дисциплин, как: «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 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
			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**	**КП**
	Экзамен	**Экзамен**	**Экзамен**

	Итого:	**184**	184

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	2	0	8
1.7	Тема 7. Строки и файлы	10	0	0	2	0	8
1.8	Тема 8. Рекурсия	10	0	0	2	0	8
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	15	1	0	4	0	10

1.15	Тема 15. Абстрактные классы и интерфейсы	15	1	0	4	0	10
1.16	Тема 16. Разработка классов	17	1	0	4	0	12
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). Оператор проверки типа переменной instanceof. Примеры программ.

Тема 3. Статические методы как функции

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.

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

Тема работы 3. Решение практических задач на последовательности.

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

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

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

Тема работы 7. Решение практических задач на файлы.

Тема работы 8. Решение практических задач на рекурсию.

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

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

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

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

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

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

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

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

3.5 Тематика курсовых проектов (курсовых работ)

Пример темы курсового проекта: «Основы объектно-ориентированного программирования. Реализация игры-симулятора «Жизнь» или студент может предложить свою тему.

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

ЭОР по данному курсу в разработке

1. Java. Базовый курс. – URL: https://stepik.org/courses/187

2. Основы информатики и программирования. – URL: https://intuit.ru/studies/courses/105/105/info

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

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 и с доступом в интернет. 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 и проектором.

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.

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.

В рамках курсового проекта студенту предлагается разработать простое оконное приложение на языке программирования Java, а также составить пояснительную записку к решению. Задание выдается в середине семестра.

В рамках экзамена студенту предлагается выполнить за ограниченное время (60 минут) два практических задания по билету.

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 балла, 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.

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 4 до 5 баллов – решение или ответ содержат ошибки; от 0 до 3 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.

Оценка за курсовой проект рассчитывается как среднее оценок за приложение и пояснительную записку с учетом ответов на вопросы по решению. 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 Текущий контроль

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 Промежуточная аттестация

Вариант экзаменационного билета:

1. Напишите программу, вводящую последовательность целых чисел и вычисляющую количество элементов, равных наибольшему.

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