Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 05.09.2026 14:03:59
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
«Информационных технологий»
_____ / Д.Г. Демидов /
«16» февраля 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Языки программирования»

Направление подготовки
10.03.01 «Информационная безопасность»

Профиль
«Безопасность компьютерных систем»

Квалификация
Бакалавр

Формы обучения
Очная

Москва, 2026 г.

Разработчик(и):

Заведующий кафедрой «Информационная безопасность»,
к.т.н., доцент  /И.В. Калуцкий/

Согласовано:

Заведующий кафедрой «Информационная безопасность»,
к.т.н., доцент  /И.В. Калуцкий/

Руководитель образовательной программы,  А.Ю. Гневшев

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
4	Учебно-методическое и информационное обеспечение	6
4.1	Основная литература	6
4.2	Дополнительная литература	7
4.3	Электронные образовательные ресурсы	7
5	Материально-техническое обеспечение	7
6	Методические рекомендации	7
6.1	Методические рекомендации для преподавателя по организации обучения	7
6.2	Методические указания для обучающихся по освоению дисциплины	8
7	Фонд оценочных средств	8
7.1	Методы контроля и оценивания результатов обучения	8
7.2	Шкала и критерии оценивания результатов обучения	8
7.3	Оценочные средства	11
7.3.1.	Список вопросов	11

1 Цели, задачи и планируемые результаты обучения по дисциплине

К **основным целям** освоения дисциплины «Языки программирования» следует отнести:

- изучение современных языков и методов разработки программного обеспечения.

К **основным задачам** освоения дисциплины «Языки программирования» следует отнести:

- освоение основных принципов и методов объектно-ориентированного программирования;
- освоение низкоуровневых возможностей языков программирования;
- освоение способов сочетания высокоуровневых и низкоуровневых методов разработки программного обеспечения на примере языка C++;
- изучение сложных программных средств разработки программного обеспечения.

Обучение по дисциплине **«Языки программирования»** направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-7. Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	**знать:** ● принципы организации и компиляции многофайловых программ на языке C++; ● принципы использования шаблонов для разработки родственных типов данных; ● принципы использования механизма наследования для разработки родственных типов данных; ● принципы использования полиморфных методов реализации алгоритмов. ● классификацию и характеристики базовых типов языка C++; ● принципы использования указателей и ссылок; ● понятие класса как типа, определяемого пользователем; ● принципы защиты данных в объектно-ориентированном программировании; ● принципы создания, инициализации и удаления объектов в объектно-ориентированном программировании; ● способы обработки исключительных ситуаций; **уметь:** ● использовать технологии объектно-ориентированного программирования для разработки программного обеспечения; ● использовать Microsoft Visual Studio или другой аналогичный редактор для разработки Windows-приложений на языке C++; ● разрабатывать программы на языке программирования C++, в том числе и с

	использованием низкоуровневых средств языка; ● разрабатывать классы. **владеть:** ● методологией и навыками решения практических задач с использованием технологий объектно-ориентированного программирования; ● методологией и навыками решения практических задач с использованием низкоуровневых средств языков программирования; ● навыками понимания кода разработанного программного обеспечения; ● терминологией объектно-ориентированного программирования, навыками профессиональной коммуникации, необходимыми при коллективной разработке программного обеспечения.

2 Место дисциплины в структуре образовательной программы

Дисциплина «Языки программирования» относится к числу профессиональных учебных дисциплин обязательной части цикла (Б.1.1) основной образовательной программы (Б.1.1.).

Изучение дисциплины опирается на знания, умения и навыки, приобретенные в предшествующих дисциплинах: «Математическая логика и теория алгоритмов в программировании».

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единицы, т.е. **216** академических часов (лабораторные занятия – 100 час, самостоятельная работа - 116 часов, форма контроля – дифференцированный зачет, экзамен) в 1, 2 семестрах.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			Семестр	Неделя семестра
1	**Аудиторные занятия**	**100**	1, 2	1-18
	В том числе:			
1.1	Лекции			
1.2	Лабораторные занятия	100	1, 2	1-18
1.3	Лабораторные занятия			
2	**Самостоятельная работа**	**116**	1, 2	1-18

3	**Промежуточная аттестация**			
	Экзамен, диф. зачет		1, 2	19-21
	Итого:	**216**		

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	Основные понятия языка программирования.	36			16		20
2	Средства описания действий.	32			16		16
3	Структурные типы данных.	36			16		20
4	Указательные типы. Динамические структуры данных.	36			16		20
5	Функциональная декомпозиция сложных программ.	36			16		20
6	Механизмы реализации объектно-ориентированного программирования в языке высокого уровня.	40			20		20
Итого		**216**			**100**		**116**

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Малов, А. В. Концепции современного программирования : учебное пособие для вузов / А. В. Малов, С. В. Родионов. — Москва : Издательство Юрайт, 2023. — 96 с. — (Высшее образование). — ISBN 978-5-534-14911-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/520338
2. Объектно-ориентированное программирование с примерами на C# : учеб. пособие / П.Б. Хорев. — М. : ФОРУМ : ИНФРА-М, 2019. — 200 с. — (Высшее образование: Бакалавриат)
3. Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для среднего профессионального образования / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2023. — 335 с. — (Профессиональное образование). — ISBN 978-5-534-05780-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/515206.

4. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/532446.
5. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2023. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/530800.

4.2 Дополнительная литература

1. Сорокин, А.А. Объектно-ориентированное программирование : учебное пособие (курс лекций) / А.А. Сорокин ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2014. – 174 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=457696 (дата обращения: 18.08.2019). – Текст : электронный.
2. Хиценко, В.П. Основы программирования : учебное пособие / В.П. Хиценко ; Министерство образования и науки Российской Федерации, Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2015. – 83 с. : схем. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=438365 (дата обращения: 18.08.2019). – Библиогр. в кн. – ISBN 978-5-7782-2706-4. – Текст : электронный.
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 227 с. — (Высшее образование). — ISBN 978-5-534-17323-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/532868.
4. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2023. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/511891.
5. Абрамян, М.Э. Введение в стандартную библиотеку шаблонов C++. Описание, примеры использования, учебные задачи: учебник по курсу «Стандартная библиотека C++» для студентов направления 02.03.02 «Фундаментальная информатика и информационные технологии» (бакалавриат) / М.Э. Абрамян ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет». – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. – 179 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=499454 (дата обращения: 18.08.2019). – Библиогр. в кн. – ISBN 978-5-9275-2374-0. – Текст : электронный.

6. Синицын, С.В. Основы разработки программного обеспечения на примере языка С / С.В. Синицын, О.И. Хлытчиев. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 212 с. : схем., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=429186 (дата обращения: 18.08.2019). – Библиогр. в кн. – Текст : электронный.
7. Программирование на языке С++ в среде Qt Creato / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк и др. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 716 с. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=428929 (дата обращения: 18.08.2019). – Библиогр. в кн. – Текст : электронный.

4.3. Электронные образовательные ресурсы

Электронный образовательный ресурс разрабатывается.

5 Материально-техническое обеспечение

Для проведения всех видов занятий необходимо презентационное оборудование (мультимедийный проектор, экран) – 1 комплект.

Для проведения лабораторных занятий необходимо наличие компьютерных классов оборудованных современной вычислительной техникой из расчета одно рабочее место на одного обучаемого.

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, задачи 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, 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 лекции.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, готовятся к экзамену, а также самостоятельно изучают отдельные темы учебной программы.

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:

- защита лабораторных работ;
- экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю):

ОПК-7 Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: •принципы организации и компиляции многофайловых программ на языке C++; •принципы использования шаблонов для разработки родственных типов данных; •принципы использования механизма наследования для разработки родственных типов данных; •принципы использования полиморфных методов реализации алгоритмов. •классификацию и характеристики базовых типов языка C++; •принципы использования указателей и ссылок; •понятие класса как типа, определяемого пользователем;	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: •принципы организации и компиляции многофайловых программ на языке C++; •принципы использования шаблонов для разработки родственных типов данных; •принципы использования механизма наследования для разработки родственных типов данных; •принципы использования полиморфных методов реализации алгоритмов. •классификацию и характеристики базовых типов языка C++; •принципы	Обучающийся демонстрирует неполное соответствие следующих знаний: •принципы организации и компиляции многофайловых программ на языке C++; •принципы использования шаблонов для разработки родственных типов данных; •принципы использования механизма наследования для разработки родственных типов данных; •принципы использования полиморфных методов реализации алгоритмов. •классификацию и характеристики базовых типов языка C++; •принципы использования указателей и ссылок;	Обучающийся демонстрирует частичное соответствие следующих знаний: •принципы организации и компиляции многофайловых программ на языке C++; •принципы использования шаблонов для разработки родственных типов данных; •принципы использования механизма наследования для разработки родственных типов данных; •принципы использования полиморфных методов реализации алгоритмов. •классификацию и характеристики базовых типов языка C++; •принципы использования	Обучающийся демонстрирует полное соответствие следующих знаний: •принципы организации и компиляции многофайловых программ на языке C++; •принципы использования шаблонов для разработки родственных типов данных; •принципы использования механизма наследования для разработки родственных типов данных; •принципы использования полиморфных методов реализации алгоритмов. •классификацию и характеристики базовых типов языка C++; •принципы использования указателей и ссылок;

•принципы защиты данных в объектно-ориентированном программировании; •принципы создания, инициализации и удаления объектов в объектно-ориентированном программировании; •способы обработки исключительных ситуаций.	использования указателей и ссылок; •понятие класса как типа, определяемого пользователем; •принципы защиты данных в объектно-ориентированном программировании; •принципы создания, инициализации и удаления объектов в объектно-ориентированном программировании; •способы обработки исключительных ситуаций.	•понятие класса как типа, определяемого пользователем; •принципы защиты данных в объектно-ориентированном программировании; •принципы создания, инициализации и удаления объектов в объектно-ориентированном программировании; •способы обработки исключительных ситуаций. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	указателей и ссылок; •понятие класса как типа, определяемого пользователем; •принципы защиты данных в объектно-ориентированном программировании; •принципы создания, инициализации и удаления объектов в объектно-ориентированном программировании; •способы обработки исключительных ситуаций, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	•понятие класса как типа, определяемого пользователем; •принципы защиты данных в объектно-ориентированном программировании; •принципы создания, инициализации и удаления объектов в объектно-ориентированном программировании; •способы обработки исключительных ситуаций, свободно оперирует приобретенными знаниями.
уметь: •использовать технологии объектно-ориентированного программирования для разработки программного обеспечения; •использовать Microsoft Visual Studio или другой аналогичный редактор для разработки Windows-приложений на языке C++; •разрабатывать программы на языке программирования C++, в том числе и с использованием низкоуровневых средств языка; •разрабатывать классы.	Обучающийся не умеет или в недостаточной степени умеет •использовать технологии объектно-ориентированного программирования для разработки программного обеспечения; •использовать Microsoft Visual Studio или другой аналогичный редактор для разработки Windows-приложений на языке C++; •разрабатывать программы на языке программирования C++, в том числе и с использованием низкоуровневых средств языка; •разрабатывать классы.	Обучающийся демонстрирует неполное соответствие следующих умений: •использовать технологии объектно-ориентированного программирования для разработки программного обеспечения; •использовать Microsoft Visual Studio или другой аналогичный редактор для разработки Windows-приложений на языке C++; •разрабатывать программы на языке программирования C++, в том числе и с использованием низкоуровневых средств языка; •разрабатывать классы. Допускаются значительные ошибки, проявляется недостаточность умений.	Обучающийся демонстрирует частичное соответствие следующих умений: •использовать технологии объектно-ориентированного программирования для разработки программного обеспечения; •использовать Microsoft Visual Studio или другой аналогичный редактор для разработки Windows-приложений на языке C++; •разрабатывать программы на языке программирования C++, в том числе и с использованием низкоуровневых средств языка; •разрабатывать классы.	Обучающийся демонстрирует полное соответствие следующих умений: •использовать технологии объектно-ориентированного программирования для разработки программного обеспечения; •использовать Microsoft Visual Studio или другой аналогичный редактор для разработки Windows-приложений на языке C++; •разрабатывать программы на языке программирования C++, в том числе и с использованием низкоуровневых средств языка; •разрабатывать классы.

			Умения освоены, но допускаются незначительные ошибки, неточности.	Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: •методологией и навыками решения практических задач с использованием технологий объектно-ориентированного программирования; •методологией и навыками решения практических задач с использованием низкоуровневых средств языков программирования; •навыками понимания кода разработанного программного обеспечения; •терминологией объектно-ориентированного программирования, навыками профессиональной коммуникации, необходимыми при коллективной разработке программного обеспечения.	Обучающийся не владеет или в недостаточной степени владеет •методологией и навыками решения практических задач с использованием технологий объектно-ориентированного программирования; •методологией и навыками решения практических задач с использованием низкоуровневых средств языков программирования; •навыками понимания кода разработанного программного обеспечения; •терминологией объектно-ориентированного программирования, навыками профессиональной коммуникации, необходимыми при коллективной разработке программного обеспечения.	Обучающийся владеет •методологией и навыками решения практических задач с использованием технологий объектно-ориентированного программирования; •методологией и навыками решения практических задач с использованием низкоуровневых средств языков программирования; •навыками понимания кода разработанного программного обеспечения; •терминологией объектно-ориентированного программирования, навыками профессиональной коммуникации, необходимыми при коллективной разработке программного обеспечения, но допускаются значительные ошибки, проявляется недостаточность владения	Обучающийся частично владеет •методологией и навыками решения практических задач с использованием технологий объектно-ориентированного программирования; •методологией и навыками решения практических задач с использованием низкоуровневых средств языков программирования; •навыками понимания кода разработанного программного обеспечения; •терминологией объектно-ориентированного программирования, навыками профессиональной коммуникации, необходимыми при коллективной разработке программного обеспечения, навыки освоены, но допускаются незначительные ошибки, неточности, затруднения.	Обучающийся в полном объеме владеет •методологией и навыками решения практических задач с использованием технологий объектно-ориентированного программирования; •методологией и навыками решения практических задач с использованием низкоуровневых средств языков программирования; •навыками понимания кода разработанного программного обеспечения; •терминологией объектно-ориентированного программирования, навыками профессиональной коммуникации, необходимыми при коллективной разработке программного обеспечения, свободно применяет полученные навыки в ситуациях повышенной сложности.

Форма промежуточной аттестации: экзамен.

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 (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1. Список вопросов

1. Стандартизация языков программирования.
2. Парадигмы программирования.
3. Способы реализации языков программирования.
4. Характеристики и свойства языков программирования.
5. История развития языков программирования.
6. Поколения языков.
7. Основные элементы языков программирования.
8. Инструментальные средства разработки программ на языках высокого уровня в
9. различных операционных системах.
10. Процесс трансляции.
11. Средства описания лексики, синтаксиса и семантики языка.
12. Алфавит, идентификаторы, служебные слова, константы, комментарии.
13. Структура программы.
14. Этапы создания исполняемой программы.
15. Типизация языка.

16. Определение типа.
17. Контроль типов.
18. Уровни типизации.
19. Эквивалентность типов.
20. Простые типы данных.
21. Переменные.
22. Классы памяти.
23. Операции, выражения, оператор присваивания.
24. Описание семантики средств описания действий.
25. Операции, выражения и операторы действия.
26. Операторы управления.
27. Операторы последовательного выполнения, условные операторы, операторы.
28. Средства стандартного ввода-вывода в консольных приложениях.
29. Определение пользовательского типа.
30. Перечислимые типы. Структурные типы данных: массивы, структуры (записи)
31. Указатели и ссылочные типы данных.
32. Статическая и динамическая память.
33. Адреса и указатели.
34. Менеджер динамической памяти.
35. Выделение и освобождение динамической памяти в процессе выполнения программы.
36. Указатели и их виды.
37. Описание типизированных указателей.
38. Работа с типизированными указателями: присваивание, разыменование, размещение динамических данных в памяти ПК.
39. Динамические массивы.
40. Списки.
41. Подпрограммы: процедуры, функции.
42. Стандартные подпрограммы.
43. Рекурсивные подпрограммы.
44. Обработка массивов и строк.
45. Реализация простейших алгоритмов и динамических структур данных на языке программирования высокого уровня.
46. Обработка исключений.
47. Основные понятия объектно-ориентированного программирования (ООП).
48. Отличия ООП от процедурного программирования.
49. Принцип инкапсуляции.
50. Понятия наследования и полиморфизма.
51. Описание абстрактных типов данных.
52. Классы и объекты, их описание и использование.
53. Данные-элементы класса.
54. Методы класса