Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 13.11.2023 16:45:08
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/
«10» мая 2022

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Языки программирования»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Профиль
«Системная и программная инженерия»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2022 г.

Разработчик(и):

ст.преподаватель /И.Н.Никишина/

Согласовано:

Руководитель образовательной программы  /А.Ю.Гневшев/

Заведующий кафедрой «Инфокогнитивные технологии»:



к.т.н., доцент /Е.А.Пухова /

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 участвовать в настройке и наладке программно-аппаратных комплексов	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
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения

	работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

2 Место дисциплины в структуре образовательной программы

Дисциплина «Языки программирования» относится к базовой части основной профессиональной образовательной программы бакалавриата.

Дисциплина «Языки программирования» взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Математическая логика и теория алгоритмов в программировании.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, т.е. 144 академических 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**	**72**	72	
	В том числе:			
1.1	Лекции			
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	72	72	
2	**Самостоятельная работа**	**72**	72	
3	**Промежуточная аттестация**			
	Курсовой проект			
	Экзамен		экзамен	
	Итого:	**144**	144	

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.	24			12		12
2	Средства описания действий.	24			12		12
3	Структурные типы данных.	24			12		12
4	Указательные типы. Динамические структуры данных.	24			12		12
5	Функциональная декомпозиция сложных программ.	24			12		12
6	Механизмы реализации объектно-ориентированного программирования в языке высокого уровня.	24			12		12
Итого		144			72		72

4 Учебно-методическое и информационное обеспечение

4.1. Основная литература

1. Синицын, С. В. Основы разработки программного обеспечения на примере языка С : учебное пособие / С. В. Синицын, О. И. Хлытчиев. — 2-е изд. — Москва : ИНТУИТ, 2016. — 211 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100345 — Режим доступа: для авториз. пользователей.
2. «Баранова И. В., Баранов С. Н., Баженова И. В., Кучунова Е. В., Толкач С. Г.Объектно-ориентированное программирование на С++» (Объектно-ориентированное программирование на С++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. — Красноярск : СФУ, 2019. — ISBN 978-5-7638-

4034-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/157572 — Режим доступа: для авториз. пользователей. — С. 2.).https://reader.lanbook.com/book/157572?lms=1e44a63afcb52611d6021fa0b8646c62#2

3. Программирование на языке C++ в среде Qt Creator : учебное пособие / Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк, О. В. Чеснокова. — 2-е изд. — Москва : ИНТУИТ, 2016. — 715 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100414 — Режим доступа: для авториз. пользователей.
4. Лямин А.В Языки программирования C/C++ : учебное пособие / А. В. Лямин, Е. Н. Череповская. — Санкт-Петербург : НИУ ИТМО, 2017 — 71 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/110458 — Режим доступа:для авториз. пользователей.

4.2. Дополнительная литература

1. Сорокин, А. А. Объектно-ориентированное программирование. LAZARUS (Free Pascal) : учебное пособие / А. А. Сорокин. — Ставрополь : СКФУ, 2014. — 216 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/155254 — Режим доступа: для авториз. пользователей.
2. Тракимус, Ю. В. Основы программирования : учебное пособие / Ю. В. Тракимус, В. П. Хиценко. — Новосибирск : НГТУ, 2020. — 66 с. — ISBN 978-5-7782-4089-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/152224 — Режим доступа: для авториз. пользователей.

4.3. Электронные образовательные ресурсы

Ссылка на электронные образовательные ресурсы
https://online.mospolytech.ru/enrol/index.php?id=9464
https://online.mospolytech.ru/enrol/index.php?id=8537

5. Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Практические занятия (семинары) и самостоятельная работа студентов должна проводиться в специализированных аудиториях с комплектом мультимедийного оборудования и/или доской для записей материалов. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

5.2 Требования к программному обеспечению

Для проведения практических занятий (семинаров) специального программного обеспечения для освоения дисциплины не требуется.

6. Методические рекомендации

6.1. Методические рекомендации для преподавателя по организации обучения

1. 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.

2. 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 аудиторные занятия, семинары и практики.

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

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

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

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

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

Критериями оценки результатов самостоятельной работы студента являются:

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

7. Фонд оценочных средств

7.1. Методы контроля и оценивания результатов обучения

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

- Экзамен;
- Курсовой проект.

7.2. Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ОПК-7. Способен участвовать в настройке и наладке программно-аппаратных комплексов				

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	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения				
ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.

программы на основе спроектированного алгоритма и проводить отладку программы, применять методы системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.		затруднения при оперировании знаниями при их переносе на новые ситуации.		

Шкала оценивания результатов промежуточной аттестации и её описание

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

Шкала оценивания	Описание
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. 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	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. 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	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. 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 достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. 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. Варианты тематик заданий для сдачи дифференцированного зачета

4. Состав языка.
5. Типы данных.
6. Структура программы.
7. Описание переменных.
8. Простейшие средства ввода-вывода.
9. Выражения.
10. Преобразования базовых типов.
11. Основные операторы.
12. Составные типы данных в стиле C
13. Функции в C++.

14. Функции библиотеки языка С.
15. Директивы препроцессора. Области действия и пространства имен.
16. Абстрактные структуры данных.
17. Динамические структуры данных.

7.3.1 Экзаменационные вопросы

1. Поколения ЯП. Классификация ЯП. Парадигмы программирования.
2. Простейшая программа на языке С. Структура программы. Комментарии.
3. Нотация в форме BNF и ее расширения.
4. Простые типы (целочисленный, вещественный, перечисляемый, пустой — void, символьный).
5. Определение именованных констант. Определение типов (typedef).
6. Переменные. Области видимости пересенных.
7. Операции и операнды. Унарные, бинарные, префиксные, постфиксные и инфиксные операции.
8. Операции и операнды. Приоритет. Ассоциативность (лево- и право-). Основные операции в С. Неявное приведение типов.
9. Условный оператор. Составная инструкция.
10. Операторы циклов (for, do...while, while). Инструкции преждевременного выхода и перехода к следующему циклу.
11. Инструкция возврата из функции. Инструкция выбора.
12. Функции. Объявление и определение.
13. Массив как набор однотипных данных. Определение. Обращение к отдельным элементам.
14. Многомерные массивы. Автоопределение внешней размерности.
15. Строки как массивы типа char. Символы как их коды. Символы конца строки в разных ОС.
16. Память. Размещение переменных в памяти. Адреса, указатели. Получение адреса (&) и разыменование (*).
17. Операции над указателями (арифметика, сравнение). Нетипизированные указатели.
18. Расположение элементов массива в памяти. Имя массива как адрес первого элемента. Операция [].
19. Указатели и многомерные массивы.
20. Статическое и динамическое выделение памяти. Динамические массивы. Функции malloc, calloc, realloc, free.
21. Основные ошибки при выделении памяти: неинициализированные указатели, висячие ссылки, утечки памяти.
22. Три способа создания многомерных динамических массивов.
23. Рекурсия. Примеры.
24. Хвостовая рекурсия. Связь с обычными циклами.
25. Предобработка (препроцессинг) исходных текстов. Основные этапы.
26. Директивы #include, #define, #undef. Макроподстановки с параметрами.
27. Условное включение текста (#ifdef, #ifndef, #if, #elif, #else, #endif). Вспомогательные директивы: #pragma, #line, #error.
28. Определение структур, теги и имена, даваемые typedef. Инициализация, доступ к элементам. Указатели и доступ к полям через указатель.
29. Объединения и перечисления. Особенности расположения в памяти и инициализации.
30. Функции для консольного ввода-вывода.
31. Файлы как абстракции, связь потоков с файлами (fopen, freopen, fclose). Разница между текстовыми и бинарными файлами.
32. Функции для работы с файлами.