Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Наумова Наталия Александровна
Должность: Ректор
Дата подписания: 24.10.2024 14:21:41
Уникальный программный ключ:
6b5279da4e034bff679172803da5b7b559fc69e2

МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ
Государственное образовательное учреждение высшего образования Московской области
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ОБЛАСТНОЙ УНИВЕРСИТЕТ
(МГОУ)
Физико-математический факультет
Кафедра вычислительной математики и методики преподавания информатики

Согласовано управлением организации и контроля качества образовательной деятельности
«08» мая 2020 г.
Начальник управления _____
/М.А. Миненкова /

Одобрено учебно-методическим советом
Протокол «12» мая 2020 г. № 07
Председатель _____
/ Г.Е. Суслин /

Рабочая программа дисциплины
Парадигмы и языки программирования

Направление подготовки
44.04.01 Педагогическое образование

Программа подготовки:
Информатика в образовании

Квалификация
Магистр

Форма обучения
Очная

Согласовано учебно-методической комиссией физико-математического факультета:
Протокол «21» мая 2020 г. № 10
Председатель УМКом _____
/Н.Н. Барабанова/

Рекомендовано кафедрой вычислительной математики и методики преподавания информатики
Протокол от «20» мая 2020 г. № 10
Зав.кафедрой _____
/М.В. Шевчук /

Мытищи
2020

Автор-составитель:

Грамаков Д. А. кандидат педагогических наук, доцент, доцент

Рабочая программа дисциплины «Парадигмы и языки программирования» составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования направление подготовки 44.04.01 Педагогическое образование, утвержденного приказом МИНОБРНАУКИ России от 22.02.18 № 126.

Дисциплина входит в часть, формируемую участниками образовательных отношений Блока 1 «Дисциплины (модули)» и является обязательной для изучения.

Год начала подготовки (по учебному плану) 2020

СОДЕРЖАНИЕ

1.ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ 4
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ 4
3. ОБЪЕМ И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ 4
4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ 6
5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ 6
6. УЧЕБНО-МЕТОДИЧЕСКОЕ И РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ 17
7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ **Ошибка! Закладка не определена.**
8. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ..... **Ошибка! Закладка не определена.**
9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ. **Ошибка! Закладка не определена.**

1.ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

1.1. Цель и задачи

Целью изучения дисциплины "Парадигмы и языки программирования " является знакомство обучающихся с современными парадигмами программирования и общими принципами построения и использования языков программирования, а также развитие навыков использования этих языков для проектирования и реализации алгоритмов решения практических задач в образовательной деятельности.

Задачи дисциплины:

1. Формирование устойчивого интереса к изучаемой дисциплине, развитие мировоззрения и творческого потенциала в области программирования, позволяющего будущему специалисту с минимальными затратами осуществлять образовательную деятельность.
2. Организация процесса обучения для развития практических навыков в области программирования при решении различных образовательных задач.
3. Планирование и обеспечение условий обучения, направленных на мобилизацию индивидуальных возможностей и ресурсов обучаемых в рамках самостоятельной работы

1.2. Планируемые результаты обучения

В результате освоения данной дисциплины у обучающихся формируются следующие компетенции:

СПК-3. Способен осуществлять научно-методическое и консультационное сопровождение процесса и результатов проектной деятельности обучающихся

СПК-4. Способен к разработке учебно-методического обеспечения для реализации образовательных программ в образовательных организациях соответствующего уровня образования

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина входит в часть, формируемую участниками образовательных отношений Блока 1 «Дисциплины (модули)» и является обязательной для изучения.

Содержание дисциплины опирается на знания обучающихся, полученные в процессе подготовки в бакалавриате в рамках освоения дисциплин: «Языки и методы программирования», «Технологии программирования для Интернета», «Алгоритмизация и основы программирования», «Практикум решения задач на ЭВМ», «Основы машинно-ориентированного программирования» а также в рамках данной программы подготовки: «Основы анализа и визуализации данных», «Современные проблемы науки и образования».

3. ОБЪЕМ И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объем дисциплины

Показатель объема дисциплины	Кол-во часов
Объем дисциплины в зачетных единицах	3
Объем дисциплины в часах	108
Контактная работа:	20,3
Лекции	4
Лабораторные занятия	14
Контактные часы на промежуточную аттестацию:	2,3
Предэкзаменационные консультации	2
Экзамен	0,3
Самостоятельная работа	78
Контроль	9,7

Форма промежуточной аттестации: экзамен в 1 семестре

3.2.Содержание дисциплины

Наименование разделов (тем) Дисциплины с кратким содержанием	Кол-во часов		
	Лекции	Практические занятия	Лабораторные занятия
Тема 1. Основные парадигмы программирования Парадигмы программирования. Различные толкования термина. «Парадигма программирования». История становления парадигм программирования. Различные подходы к классификации парадигм программирования. Основные парадигмы программирования. Языки и парадигмы. Мультипарадигмальные языки программирования. Понятие алгоритма и свойства алгоритмов. Представление алгоритмов. Машина Тьюринга. **Понятие структурного (процедурного) программирования**. Архитектура фон Неймана. Принципы фон Неймана. Безусловный переход goto (jump). Языки структурного (процедурного) программирования. Теорема Бёма – Якопини. Принципы структурного программирования. Управляющие конструкции. Процедуры и функции. Условные операторы. Варианты ветвлений. Циклы с постусловием. Цикл со счетчиком.Цикл с предусловием. Цикл для работы с коллекциями. **Объектно-ориентированное программирование.** Классы. Методы. История объектно-ориентированного программирования. Объект. Абстрагирование. Инкапсуляция и сокрытие. Наследование. Полиморфизм. Принципы объектно-ориентированного программирования по Алану Кею. UML-диаграмма класса. Видимость в UML. **Функциональное программирование.** Модель вычисления в функциональных языках. Функции высших порядков. Карринг. Чистые функции Рекурсия. Требования к рекурсии. Рекурсивные алгоритмы. Подходы к вычислению аргументов. **Языки функционального программирования Параллельное программирование**. Параллельные вычисления. Подходы к разработке параллельных программ. Технологии параллельного программирования. **Компонентно-ориентированное программирование**. Реализация в Oberon, JavaBeans и .NET. **Прототипное программирование.** Реализация в Javascript и Lua. Сравнение класс-ориентированного и прототипного программирования. **Технология Markdown**. Технологии R Markdown. **Событийно-ориентированное программирование**. Аспекты событий. Области применения. Языки событийно-ориентированного программирования. **Визуальное программирование**. Среды визуального программирования.	2		4
Тема 2. Языки программирования. **Язык программирования C#.** Основы языка. Базовые понятия. Подходы к реализации различных парадигм программирования. **Язык функционального программирования F#.** Основы языка. Подходы к реализации функциональной парадигмы программирования. Среда программирования. **Язык программирования Python**. Основы языка. Базовые понятия. Подходы к реализации различных парадигм программирования. Среда программирования. **Язык программирования R**.	2		10

Основы языка. Базовые понятия. Подходы к реализации различных парадигм программирования. Среда программирования. **Язык программирования JavaScript**. Основы языка. Базовые понятия. Подходы к реализации объектно-ориентированноей и прототипной парадигм программирования. Среда программирования. **Язык программирования TypeScript**. Основы языка. Базовые понятия. Подходы к реализации объектно-ориентированноей и прототипной парадигм программирования. Среда программирования **Язык XML.**			
ИТОГО	4		14

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Тема	**Изучаемые вопросы**	**Кол-во час.**	**Формы самостоятельной работы**	**Методическое обеспечение**	**Форма отчета**
Тема 1. Основные парадигмы программирования	Парадигмы программирования: структурная, объектно-ориентированная, функциональная, логическая.	24	Изучение литературы, документов в Интернете, текстов программ	Учебно-методическое обеспечение	Домашнее задание
Тема 2. Языки программирования.	Языки и среды программирования: C#, F#, Python, R, JavaScript, TypeScript, XML.	54	Изучение литературы, документов в Интернете, текстов программ на различных языках программирования. Решение задач	Учебно-методическое обеспечение	Домашнее задание
ИТОГО		78			

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

5.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Код и наименование компетенции	**Этапы формирования**

Код и наименование компетенции	Этапы формирования
СПК-3. Способен осуществлять научно-методическое и консультационное сопровождение процесса и результатов проектной деятельности обучающихся	1. Работа на учебных занятиях 2. Самостоятельная работа
СПК-4. Способен к разработке учебно-методического обеспечения для реализации образовательных программ в образовательных организациях соответствующего уровня образования	1. Работа на учебных занятиях 2. Самостоятельная работа

5.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Оцениваемые компетенции	Уровень сформированности	Этап формирования	Описание показателей	Критерии оценивания	Шкала Оценивания
СПК-3	Пороговый	1. Работа на учебных занятиях 2. Самостоятельная работа	Знать: - технологии и подходы для поддержки процесса и результатов проектной деятельности обучающихся Уметь - поддерживать процесс и результаты проектной деятельности учащихся	Тестирование Домашние задания, лабораторные работы	Шкала оценивания домашнего задания Шкала оценивания тестирования Шкала оценивания лабораторных работ
	Продвинутый	1. Работа на учебных занятиях 2. Самостоятельная работа	Знать: - технологии и подходы для поддержки процесса и результатов проектной деятельности обучающихся Уметь: - поддерживать процесс и результаты проектной деятельности учащихся Владеть: - навыками и технологиями поддержки процесса и результатов проектной деятельности обучающихся	Тестирование Домашние задания, лабораторные работы Домашние задания, лабораторные работы	Шкала оценивания домашнего задания Шкала оценивания тестирования Шкала оценивания лабораторных работ
СПК-4	Пороговый	1.Работа на учебных занятиях 2. Самостоятельная работа	Знать: - методы и средства разработки учебно-методического обеспечения образовательных программ Уметь: - сопровождать разработку	Тестирование Домашние	Шкала оценивания домашнего задания Шкала оценивания тестирования Шкала

			учебно-методического обеспечения образовательных программ	задания, лабораторные работы	оценивания лабораторных работ
	Продвинутый	1.Работа на учебных занятиях 2. Самостоятельная работа	Знать: - методы и средства разработки учебно-методического обеспечения образовательных программ Уметь: - сопровождать разработку учебно-методического обеспечения образовательных программ Владеть: - навыками разработки учебно-методического обеспечения для реализации образовательных программ	Тестирование Домашние задания, лабораторные работы Домашние задания, лабораторные работы	Шкала оценивания домашнего задания Шкала оценивания тестирования Шкала оценивания лабораторных работ

Шкала оценивания домашнего задания

Критерии оценивания	Баллы
Аккуратность и полнота выполнения всех пунктов задания.	0-1
Понимание логики выполнения задания и значения полученных результатов	0-1
Правильность и самостоятельность написания кода и комментариев	0-1

Шкала оценивания лабораторных работ

Критерии оценивания	Баллы
Аккуратность и полнота выполнения всех пунктов задания	0-1
Понимание логики выполнения задания и значения полученных результатов	0-1
Правильность и самостоятельность написания кода и комментариев	0-1

Шкала оценивания тестирования

Уровень оценивания	Критерии оценивания	Баллы
Выполнение теста	Тест выполнено полностью (75%-100%)	10

	Тест выполнен частично (50%-74%)	5
	Тест не выполнен или выполнено менее 50% (0%-49%)	0

5.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Примеры тестовых заданий

1. Какие из перечисленных языков программирования основаны на функциональном подходе?
   а) SML и ProLog;
   б) LISP и ProLog;
   в) SMLnLISP(+).

2. Что отличает императивные языки программирования от декларативных?
   а) степень зависимости от среды реализации;
   б) стиль программирования (+);
   в) структура программы (+).

3. В чем состоит особенность языков объектно-ориентированного программирования?
   а) этот класс языков основан па сценариях;
   б) этот класс языков концептуально близок к любой предметной области (+);
   в) этот класс языков является наиболее машинно-независимым.

4. Какие из перечисленных языков программирования основаны на объектно-ориентированном подходе?
   а) C# и SML;
   б) C# и C++(+);
   в) C# и ProLog.

5. Что отличает ранние языки программирования от языков созданных в последнее время?
   а) степень зависимости от среды реализации (+);
   б) программы на ранних языках более наглядны;
   в) существенных различий пет.

6. В чем состоит особенность языков логического программирования?
   а) этот класс языков основан на функциях;
   б) моделирует правила (+);
   в) является наиболее машинно-независимым.

7. Какие из перечисленных языков программирования основаны на структурном подходе?
   а) C# и ProLog;
   б) C# и Fortran (+);
   в) C# и SML.

8. Что отличает объектно-ориентированный подход к программированию от компонентно-ориентированного?

а) степень зависимости от среды реализации;
б) стиль программирования;
в) структура программы (+).

9. В чем состоит особенность языков функционального программирования?
а) этот класс языков основан на сценариях;
б) концептуально близок к любой предметной области;
в) легко формализуем математически (+).

10. Что понимается под термином «объект»?
а) собственно сущность реального мира;
б) абстрактное представление сущности реального мира;
в) математическое представление сущности реального мира (+).

11. Каково соотношение понятий «объект» и «класс»?
а) понятие «объект» является более общим;
б) понятие «класс» является более общим (+);
в) эти понятия несопоставимы.

12. Что понимается под термином «свойство»?
а) абстрактная характеристика объекта;
б) математическая характеристика объекта;
в) пропозициональная операция над объектами (+).

13. Чем является объект по отношению к классу?
а) экземпляром (+);
б) типом;
в) реализацией.

14. Что понимается под термином «метод»?
а) алгоритм решения задачи;
б) операция над объектами (+);
в) подход к программированию.

15. На каких математических теориях основана концептуальная модель?
а) на типизированном лямбда-исчислении и теории решеток (+);
б) на бестиповом лямбда-исчислении и теории решеток;
в) на теории категорий и теории решеток.

16. В чем состоит основное назначение семантики?
а) в формализации вида и формы конструкций языка;
б) в формализации значения конструкций языка (+);
в) в формализации абстрактной машины для реализации языка.

17. Какая из теорий не является формализацией семантики?
а) аксиоматический метод Хоара;
б) формы Бэкуса — Наура (+);
в) метод индуктивных утверждений Р. Флойда.

18. Каковы виды семантик, ориентированные на интерпретацию?
а) операционные, препозиционные, денотационные (+);
б) операционные, композиционные, денотационные;

в) операционные, аппликативные, денотационные.

19. Что понимается под семантикой?
    а) модель интерпретации абстрактного синтаксиса (+);
    б) модель реализации языка программирования;
    в) модель предметной области.

20. Какая формализация относится к денотационным семантикам?
    а) теория вычислений Д. Скотта (+);
    б) аксиоматический метод Ч. Хоара;
    в) абстрактная машина П. Лендина.

21. Что из перечисленного является формализацией семантики?
    а) теория вычислений Д. Скотта (+);
    б) комбинаторная логика Х. Карри;
    в) абстрактная машина П. Лендина (+).

22. На что ориентированы основные подходы к семантике?
    а) на компиляцию (+);
    б) на интерпретацию (+);
    в) на корректность типизации.

23. какая формализация относится к операционным семантикам?
    а) теория вычислений Д. Скотта;
    б) аксиоматический метод Ч. Хоара;
    в) абстрактная машина П. Лендина (+).

24. Какие из перечисленных языков программирования основаны на функциональном подходе?
    а) SML и ProLog;
    б) LISP и ProLog;
    в) SML и LISP(+).

25. В чем состоит преимущество функционального подхода к программированию перед другими подходами?
    а) близость к предметной области;
    б) прозрачность реализации рекурсии (+);
    в) высокая вычислительная эффективность.

26. В чем состоит недостаток языков функционального программирования?
    а) высокая степень машинной независимости;
    б) нелинейная структура программы (+);
    в) узкая проблемная ориентированность.

27. Каково соотношение понятий функции в математике и программировании?
    а) это несопоставимые понятия;
    б) это эквивалентные понятия;
    в) математические функции моделируют функции в программировании (+).

28. Что отличает лямбда-исчисление от булевой логики?
    а) лямбда-исчисление адекватно формализует процесс программирования (+);
    б) булева логика более наглядно формализует процесс программирования;

в) существенных различий нет.

29. В чем состоит особенность языка функционального программирования F#?
а) это бестиповый язык программирования;
б) это язык программирования с полиморфной типизацией (+);
в) это язык программирования с параметрическим полиморфизмом.

Пример домашнего задания

Задание 1. Расположить 8 ферзей отдельно друг от друга на шахматной доске.

Задача «8 ферзей» на шахматной доске соответствует поиску размещения 8 ферзей, так чтобы ни один ферзь (королева) не мог атаковать никакого другого ферзя. Это означает, что

1. Никакие две королевы не могут быть в одном столбце.
2. Никакие две королевы не могут быть в одном ряду.
3. Никакие две королевы не могут быть на одной диагонали.

Найти решения данной задачи?

Задание2 : изменить код EightQueens из задания 1 так, чтобы он принимал в качестве дополнительного аргумента количество решений, которые необходимо найти и вывести их на печать, предполагая, что их много.

Пример лабораторной работы по дисциплине «Парадигмы и языки программирования»:

Лабораторная работа №5

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ (ИТЕРАЦИИ И КОНВЕЙЕРЫ)

Цель: познакомиться с особенностями функционального программирования. Научиться применять функциональное программирования с использованием пакета purrr. Решить задания в соответствующем стиле программирования. Составить отчет.

Теоретические сведения

Язык R поддерживает функциональную парадигму программирования, однако для лучшей ее реализации, и для увеличения функционала, рекомендуется использовать пакет purrr, кроме того, необходимо также скачать и подключить пакет repurrrsive, содержащий наборы данных для выполнения задания №1.

Итерации

Итерация - организация обработки данных, при которой действия повторяются многократно. В программировании итерации чаще рассматриваются в качестве элементов структурного программирования, а именно как единичный шаг выполнения цикла. На практике итерации также важны и в функциональном программировании, например в качестве итерабельного процесса можно рассматривать применение одной и той же функции к разным элементам. Рассмотрим пример выполнения кода, написанного в структурном и функциональном стиле над однотипным набором данных.

```
#  Создание тестового списка iris_list <- as.list(iris[1:4])

#  Реализация в структурном стиле iris_mean <- list() for (i in seq_along(iris_list)) {
iris_mean[[i]] <- mean(iris_list[[i]])
}
```

```
#  Реализация в функциональном стиле через map iris_mean <- map(iris_list, mean)
```

Рисунок 1 - Сравнение функциональной и структурной реализации

У функции map есть два варианта реализации (рисунок 2).

```
#  Реализация map без указания точки входа в функцию iris_mean <- map(iris_list, mean)

#  Реализация map с указанием точки входа в функцию iris_mean <- map(iris_list, ~
mean(.x))
```

Рисунок 2 - Варианты синтаксиса map

Второй вариант реализации позволяет показать место передачи элемента листа (“.x”) через знак тильда (“~”). Второй способ является предпочтительнее для реализации.

Вариации функции map. map_*

Функция map на выходе выдает список, аналогичной структуры, однако часто необходимо сразу преобразовать вывод в некоторый вариант, для этого используются производных от функции map:

```
map(.x, .f, ...)
map_if(.x, .p, .f, ...)
map_at(.x, .at, .f, ...)
map_lgl(.x, .f, ...)
map_chr(.x, .f, ...)
map_int(.x, .f, ...)

map_dbl(.x, .f, ...)
map_dfr(.x, .f, ..., .id = NULL)
map_dfc(.x, .f, ...)
walk(.x, .f, ...)
```

```
#  Создание именованного числового вектора с помощью map_dbl iris_mean <-
map_dbl(iris_list, ~ mean(.x))
```

Рисунок 3 - Создание именованного числового вектора с помощью map_dbl

Конвейеры (pipeline)

Конвейер - инструмент для передачи значения исходной функции в последующую. Конвейер представляет типичный элемент функционального программирования. В языке R конвейер имеет вид %>%. Синтаксис конвейера представлен на рисунке 4.

```
function_before() %>% function_after()
```

Рисунок 4 - Синтаксис конвейера

Практическая часть

Задание 1. Используя тестовые данные пакета repurrrsive выполните следующее задание. Создайте именованный список аналогичный по структуре списку sw_films, для установления имени полезно использовать функцию set_names пакета purrr. В качестве имени элементов списка необходимо использовать соответствующие название фильмов (обратите внимание, что обращаться к элементам списка можно используя как индекс, так и название элемента). Выполните задание в функциональном стиле.

Задание 2. Используя документацию пакета purrr опишите отличия и особенности функций семейства map_*. Приведите примеры реализации с использование различных тестовых данных. Данные можно брать из пакета datasets или создав свои тестовые наборы. Для просмотра данных из пакета datasets выполните код library(help = "datasets")

Примерный список вопросов к экзамену

1. Парадигмы программирования. Различные толкования термина «Парадигма программирования». История становления парадигм программирования
2. Различные подходы к классификации парадигм программирования. Основные парадигмы программирования. Языки и парадигмы. Языки программирования с поддержкой множества парадигм.
3. Понятие алгоритма и свойства алгоритмов. Представление алгоритмов. Машина Тьюринга.
4. Понятие процедурного программирования. Архитектура фон Неймана. Принципы фон Неймана. Безусловный переход goto (jump). Языки процедурного программирования
5. Теорема Бёма - Якопини. Принципы структурного программирования
6. Управляющие конструкции. Процедуры и функции. Условные операторы. Варианты ветвлений. Циклы с постусловием. Цикл со счетчиком. Цикл с предусловием (в одном из языков программирования)
7. Объектно-ориентированное программирование. Классы. Методы. История объектно-ориентированного программирования.
8. Объект. Абстрагирование. Инкапсуляция и сокрытие. Наследование. Полиморфизм на примере одного из языков.
9. Принципы объектно-ориентированного программирования по Алану Кею. UML-диаграмма класса. Видимость в UML
10. Векторные вычисления. Цели векторизации. Реализация в языках программирования
11. Функциональное программирование. Модель вычисления в функциональных языках. Функции высших порядков. Карринг. Чистые функции
12. Рекурсия. Требования к рекурсии. Рекурсивные алгоритмы. Подходы к вычислению аргументов. Языки функционального программирования
13. Параллельное программирование. Параллельные вычисления. Подходы к разработке параллельных программ. Технологии параллельного программирования
14. Технология Markdown. Технологии R Markdown, knitr, pandoc
15. Событийно-ориентированное программирование. Аспекты событий. Области применения. Языки событийно-ориентированного программирования
16. Визуальное программирование. Среды визуального программирования. Реализация в Scratch и App Inventor
17. Появление языков программирования. Машинный код. Первые процедурные языки высокого уровня

18. Инструкция goto. Инструкция break без метки. Инструкция break с меткой
19. Try/catch. Блок finaly в обработке исключений. Блок else в обработке исключений
20. Компонентно-ориентированное программирование. Реализация в Oberon, JavaBeans и .NET
21. Прототипное программирование. Реализация в Javascript и Lua. Сравнение класс-ориентированного и прототипного программирования
22. Декларативное программирование. Примеры реализации.
23. Реактивное программирование. Функционально-реактивный стиль программирования. Реализация в ELM
24. Сравнительный анализ языков программирования
25. Общая характеристика языков программирования

5.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Для сдачи экзамена по дисциплине необходимо выполнить все требуемые лабораторные работы. Существенным моментом является посещаемость занятий (в случае пропусков занятий предполагается более подробный опрос по темам пропущенных занятий). На экзамен выносится материал, излагаемый в лекционном курсе и рассматриваемый на лабораторных занятиях. Для сдачи экзамена надо правильно ответить на два вопроса билета и решить предложенную задачу, а также ответить на несколько поставленных дополнительных вопросов. В затруднительных ситуациях (в отдельных случаях) допускается на экзамене воспользоваться записью материалов лекций и семинаров в присутствии преподавателя. При этом преподаватель может убедиться, в какой степени студент ориентируется в «своих» материалах, и по ряду дополнительных вопросов (по тетради) решить вопрос об уровне выставляемых баллов для экзамена.

Шкала оценивания на экзамене

Критерии оценивания	Баллы
Ставится, если студент обнаруживает всестороннее, систематическое и глубокое знание основных парадигм программирования, базовых понятий языков программирования, связанных с той или иной парадигмой. Выполнил все предложенные лабораторные и самостоятельные работы, вовремя сдал отчеты по ним. При ответах по теоретическим вопросам на экзамене показал умение четко излагать изученный материал, демонстрируя взаимосвязь основных понятий дисциплины и умение оперировать ими; проявляя творческие	30

Критерии оценивания	Баллы
способности в понимании, изложении и использовании учебного материала. Правильно решил задачу, предложенную к теоретическим вопросам, показав полное понимание каждой конструкции написанной программы и понимание реализованного алгоритма решения. Показал знакомство с дополнительной литературой, рекомендованной программой. В качестве дополнительного критерия выставления оценки студент должен иметь не менее 35-40 баллов за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу.	
Ставится, если студент, обнаруживает полное знание основных парадигм программирования, базовых понятий языков программирования, связанных с той или иной парадигмой. Выполнил не менее 80 % предложенных лабораторных и самостоятельных работ, имеет незначительные нарушения по срокам сдачи отчетов. При ответах по теоретическим вопросам на экзамене по одному из вопросов не показал умения четко излагать изученный материал, демонстрируя взаимосвязь основных понятий дисциплины и умение оперировать ими. Правильно решил задачу, предложенную к теоретическим вопросам, но не показал полного понимания каждой конструкции написанной программы и не объяснил суть реализованного алгоритма решения. Не знакомился с дополнительной литературой, рекомендованной программой. В качестве дополнительного критерия выставления оценки студент должен иметь не менее 25-30 баллов за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу.	20
Ставится, если студент обнаруживает знание основных парадигм программирования, базовых понятий языков программирования, связанных с той или иной парадигмой. Выполнил не менее 60% предложенных лабораторных и самостоятельных работ, имеет незначительные нарушения по срокам сдачи отчетов. При ответе на вопросы билета не предоставил правильного ответа на один из вопросов, но при этом по другому вопросу показал умения излагать изученный материал, демонстрируя его понимание. Правильно решил задачу, предложенную к теоретическим вопросам, но не показал полного понимания каждой конструкции написанной программы и не объяснил суть реализованного алгоритма решения. В случае, когда предложенная задача не решена, но на теоретические вопросы студент ответил. Не знакомился с дополнительной литературой, рекомендованной программой. В качестве дополнительного критерия выставления оценки студент должен иметь не менее 15-20 баллов за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу.	10
Ставится в том случае, если студент на экзамене воспользовался запрещенными средствами списывая (смартфон, компьютер, бумажные носители) или не выполнил два из предложенных трех заданий билета. В качестве дополнительного критерия выставления оценки студент имеет за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу менее 15 баллов.	0

Итоговая шкала оценивания результатов освоения дисциплины

Итоговая оценка по дисциплине формируется из суммы баллов по результатам текущего контроля и промежуточной аттестации и выставляется в соответствии с приведенной ниже таблицей.

Оценка по 100-балльной системе	Оценка по традиционной системе
81 – 100	5 «отлично»
61 - 80	4 «хорошо»
41 - 60	3 «удовлетворительно»
0 - 40	2 «неудовлетворительно»

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Залогова, Л.А. Основы объектно-ориентированного программирования на базе языка C#: учеб.пособие / Л. А. Залогова. - СПб. : Лань, 2018. - 192с. – Текст: непосредственный.
2. Тюкачев, Н.А. C#. Основы программирования+CD: учеб.пособие / Н. А. Тюкачев, В. Г. Хлебостроев. - 3-е изд.,стереотип. - СПб. : Лань, 2018. - 272с. – Текст:непосредственный.
3. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451488 (дата обращения: 02.08.2020).
4. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2020. — 206 с. — (Высшее образование). — ISBN 978-5-534-00849-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451429 (дата обращения: 02.08.2020).

6.2. Дополнительная литература

1. Конова, Е.А. Алгоритмы и программы. Язык C++: учеб.пособие для вузов / Е. А. Конова, Г. А. Поллак. - 4-е изд.,стереотип. - СПб. : Лань, 2019. - 384с. – Текст: непосредственный.
2. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/450832 (дата обращения: 02.08.2020).
3. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451972 (дата обращения: 02.08.2020).
4. Подбельский, В. В. Программирование. Базовый курс C# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/456697 (дата обращения: 02.08.2020).

5. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2020. — 137 с. — (Высшее образование). — ISBN 978-5-534-07834-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/452333 (дата обращения: 02.08.2020).
6. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/119661 (дата обращения: 03.08.2020). — Режим доступа: для авториз. пользователей.
7. Диков, А.В. Веб-технологии HTML и CSS : учебное пособие / А.В. Диков. - 2-е изд. - Москва : Директ-Медиа, 2012. - 78 с. : ил.,табл., схем. - URL: http://biblioclub.ru/index.php?page=book&id=96968 (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Университетская библиотека онлайн» . — Текст : электронный
8. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие / Л. А. Залогова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2020. — 192 с. — ISBN 978-5-8114-4757-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/126160 (дата обращения: 12.10.2020). — Режим доступа: для авториз. пользователей.
9. Конова, Е. А. Алгоритмы и программы. Язык C++ : учебное пособие / Е. А. Конова, Г. А. Поллак. — 5-е изд., стер. — Санкт-Петербург : Лань, 2020. — 384 с. — ISBN 978-5-8114-5431-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140730 (дата обращения: 12.10.2020). — Режим доступа: для авториз. пользователей.
10. Солдатенко, И.С. Практическое введение в язык программирования Си : учебное пособие / И.С. Солдатенко, И.В. Попов. — Санкт-Петербург : Лань, 2018. — 132 с. — ISBN 978-5-8114-3150-2. — URL: https://e.lanbook.com/book/109619 (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Лань» . — Текст : электронный.
11. Хоган, Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган; [пер. с англ. Е. Матвеева]. - 2-е изд. - СПб.: Питер, 2014. - 320 с.: ил. - (Библиотека программиста).
12. Буйначев С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91 с.
13. МакГрат Майк. Программирование для начинающих: [перевод с англ. М.А. Райтмана] /Майк МакГрат. – Москва : Эксмо, 2015. – 192 с. – (Программирование для начинающих).
14. Чан Уэсли. Python: создание приложений. Библиотека профессионала, 3-е изд. Пер. с англ. - М. : ООО "И.Д. Вильяме", 2015. - 816 с. : ил. - Парал. тит. англ .

15. Васильев А. Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
16. Чан Уэсли. Python: создание приложений. Библиотека профессионала, 3-е изд. Пер. с англ. - М. : ООО "И.Д. Вильяме", 2015. - 816 с. : ил. - Парал. тит. англ .
17. Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).
18. Васильев А. Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
19. Бриггс Джейсон. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.
20. Буйначев С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91, [1] с.
21. МакГрат Майк. Программирование для начинающих: [перевод с англ. М.А. Райтмана] /Майк МакГрат. – Москва : Эксмо, 2015. – 192 с. – (Программирование для начинающих).
22. Саммерфилд М., Python на практике / Марк Саммерфилд - М. : ДМК Пресс, 2014. - 338 с. - ISBN 978-5-97060-095-5 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970600955.html (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
23. Маккинли У., Python и анализ данных / Уэс Маккинли - М. : ДМК Пресс, 2015. - 482 с. - ISBN 978-5-97060-315-4 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970603154.html (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
24. Лучано Рамальо, Python. К вершинам мастерства / Лучано Рамальо - М. : ДМК Пресс, 2016. - 768 с. - ISBN 978-5-97060-384-0 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : https://www.studentlibrary.ru/book/ISBN9785970603840.html (дата обращения: 12.10.2020). - Режим доступа : по подписке.
25. Мишенин А.И., Сборник задач по программированию : учеб. пособие / А.И. Мишенин. - М. : Финансы и статистика, 2014. - 224 с. - ISBN 978-5-279-03300-3 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785279033003.html. (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
26. Ошероув Р., Искусство автономного тестирования с примерами на C#. / Ошероув Р. - 2-е издание. - М. : ДМК Пресс, 2014. - 360 с. - ISBN 978-5-94074-945-5 - URL: http://www.studentlibrary.ru/book/ISBN9785940749455.html (дата

обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный

27. Робисон У., C# без лишних слов / Робисон У. ; Пер. с англ. - М. : ДМК Пресс, 2008. - 352 с. (Серия "Для программистов") - ISBN 5-94074-177-0 - URL: http://www.studentlibrary.ru/book/ISBN5940741770.html (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный

28. Тейт Брюс, Семь языков за семь недель. Практическое руководство по изучению языков программирования / Тейт Брюс - М. : ДМК Пресс, 2014. - 384 с. - ISBN 978-5-94074-539-6 - URL: http://www.studentlibrary.ru/book/ISBN9785940745396.html (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный

29. Кингсли-Хью Э., JavaScript в примерах / Кингсли-Хью Э., Кингсли-Хью К. ; Пер. с англ. - М. : ДМК Пресс, 2009. - 272 с. - ("Для программистов"). - ISBN 978-5-94074-668-3 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785940746683.html (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный

6.3. Ресурсы информационно-телекоммуникационной сети «Интернет»

1. Документация по Python [Электронный ресурс] – URL: https://docs.python.org /3/
2. Электронная библиотечная система ZNANIUM.COM http://znanium.com/
3. Электронная библиотечная система «Университетская библиотека онлайн» http://biblioclub.ru/index.php
4. Интерактивные онлайн-курсы по HTML и CSS – https://htmlacademy.ru/
5. Справочник по HTML – http://htmlbook.ru/
6. Онлайн-учебник по Javascript http://javascript.ru/
7. Интернет-портал образовательных ресурсов по ИТ - http://www.intuit.ru
8. Интернет-портал с ресурсами по программным продуктам компании Microsoft http://www.msdn.ru
9. Интернет-портал со статьями по алгоритмике и программированию - http://algolist.manual.ru/
10. Электронная библиотека по техническим наукам - http://techlibrary.ru

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

1. Методические рекомендации по проведению лабораторных и практических занятий.
2. Методические рекомендации по выполнению курсовой работы.

8. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Лицензионное программное обеспечение:
Microsoft Windows
Microsoft Office
Kaspersky Endpoint Security

Информационные справочные системы:
Система ГАРАНТ
Система «КонсультантПлюс»

Профессиональные базы данных
fgosvo.ru
pravo.gov.ru
www.edu.ru

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя:

- учебные аудитории для проведения занятий лекционного и семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, укомплектованные учебной мебелью, доской, демонстрационным оборудованием.
- помещения для самостоятельной работы, укомплектованные учебной мебелью, персональными компьютерами с подключением к сети Интернет и обеспечением доступа к электронным библиотекам и в электронную информационно-образовательную среду МГОУ;
- помещения для хранения и профилактического обслуживания учебного оборудования, укомплектованные мебелью (шкафы/стеллажи), наборами демонстрационного оборудования и учебно-наглядными пособиями;
- лаборатория, оснащенная лабораторным оборудованием: комплект учебной мебели, проектор, проекционная доска, персональные компьютеры с подключением к сети Интернет и обеспечением доступа к электронным библиотекам и в электронную информационно-образовательную среду МГОУ.