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)

Факультет физико-математический

Кафедра вычислительной математики и методики преподавания информатики

УТВЕРЖДЕН на заседании кафедры
Протокол «20» мая 2020 г. № 10

Зав. кафедрой _____
/Шевчук М.В./

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Парадигмы и языки программирования

Направление подготовки

44.04.01 Педагогическое образование

Мытищи
2020

Автор-составитель:
Грамаков Дмитрий Анатольевич,
кандидат педагогических наук, доцент, доцент кафедры вычислительной математики и методики преподавания информатики

Фонд оценочных средств по дисциплине «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 22.02.2018 г. № 126.

Дисциплина входит в вариативную часть блока 1 Дисциплины (модули) и является обязательной для изучения.

Год начала подготовки 2020

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

Изучение дисциплины «Парадигмы и языки программирования» позволяет сформировать у бакалавров следующие компетенции, необходимые для педагогической, культурно-просветительской и научно-исследовательской деятельности:

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 (лекции, лабораторные занятия) 2. Самостоятельная работа (выполнение домашних заданий, лабораторных работ)
СПК-4: «Способен к разработке учебно-методического обеспечения для реализации образовательных программ в образовательных организациях соответствующего уровня образования.»	1. Работа на учебных занятиях (лекции, лабораторные занятия) 2. Самостоятельная работа (выполнение домашних заданий, лабораторных работ)

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 (лекции, лабораторные занятия) 2. Самостоятельная работа (выполнение домашних заданий, лабораторных работ)	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 (лекции, лабораторные занятия) 2. Самостоятельная работа (выполнение домашних заданий, лабораторных работ)	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	Знать:	Тестирование	Шкала

		занятиях (лекции, лабораторные занятия) 2. Самостоятельная работа (выполнение домашних заданий, лабораторных работ)	- методы и средства разработки учебно-методического обеспечения образовательных программ Уметь: - сопровождать разработку учебно-методического обеспечения образовательных программ	Домашние задания, лабораторные работы	оценивания домашнего задания Шкала оценивания тестирования Шкала оценивания лабораторных работ
	Продвинутый	1.Работа на учебных занятиях (лекции, лабораторные занятия) 2. Самостоятельная работа (выполнение домашних заданий, лабораторных работ)	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

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

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

Оценивание степени освоения обучающимися дисциплины осуществляется на основе «Положение о балльно-рейтинговой системе оценки успеваемости студентов МГОУ».

Шкала соответствия рейтинговых оценок пятибалльным оценкам:

Оценка по 5-балльной системе		Оценка по 100-балльной системе
5	отлично	81 – 100
4	хорошо	61 - 80
3	удовлетворительно	41 - 60
2	неудовлетворительно	21 - 40
1	необходимо повторное изучение	0 - 20

В зачетно-экзаменационную ведомость и зачетную книжку выставляются оценки по пятибалльной шкале и рейтинговые оценки в баллах.

При получении студентом на экзамене неудовлетворительной оценки в ведомость выставляется рейтинговая оценка в баллах (<40 баллов), соответствующая фактическим знаниям (ответу) студента.

Общее количество баллов по дисциплине – 100 баллов.

Максимальное количество баллов, которое можно набрать в течение семестра за посещаемость, выполнение лабораторных работ, домашних заданий и тестирование – 40 баллов.

За посещение лекционных занятий и написание конспектов обучающийся может набрать максимально 3 балла.

За выполнение домашних заданий обучающийся может набрать максимально 20 баллов (8 заданий по 2,5 балла).

За выполнение лабораторных работ обучающийся может набрать максимально 7 баллов (7 работ по 1 баллу).

За тестирование обучающийся может набрать максимально 10 баллов (20 тестовых вопросов по 0,5 балла за каждый).

Обучающийся, набравший 15 балл и более, допускается к экзамену. Максимальная сумма баллов, которые обучающийся может набрать при сдаче экзамена, составляет 60 баллов.

Для сдачи экзамена по дисциплине необходимо выполнить все требуемые лабораторные работы (получить допуск к экзамену у преподавателя, проводившего 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 о уровне выставляемых баллов для экзамена.

При пересдаче экзамена используется следующее правило для формирования рейтинговой оценки:

- 1-я пересдача – фактическая рейтинговая оценка, полученная студентом за ответ, минус 10 баллов;
- 2-я пересдача – фактическая рейтинговая оценка, полученная студентом за ответ, минус 10 баллов.

Учет посещаемости лекционных и лабораторных занятий осуществляется по ведомости, представленной ниже в форме таблицы.

Московский государственный областной университет
Ведомость учета посещения
Физико-математический факультет

Направление подготовки: 44.04.01 - Педагогическое образование
Профиль подготовки: Информатика в образовании
Дисциплина: Парадигмы и языки программирования
Группа: 1
Преподаватель: Грамаков Д.А.

№ п/п	Фамилия И.О.	Посещение занятий								Итого
		1	2						9	
1.	Иванов И.И.	+	+						+	2
2.	Петров П.П.	-	+						+	1

Московский государственный областной университет
Ведомость учета текущей успеваемости
Физико-математический факультет

Направление подготовки: 44.04.01 - Педагогическое образование
Профиль подготовки: Информатика в образовании
Дисциплина: Парадигмы и языки программирования
Группа: 1
Преподаватель: Грамаков Д.А.

№ п/п	Ф. И.О.	Сумма баллов, набранных в семестре					Общая сумма баллов (макс. 100)	Итоговая оценка		Подпись преподавателя
		Посещ. до 3 баллов	Лаб. работы до 7 баллов	Домашние задания до 20 баллов	Тестирование до 10 баллов	Экзамен до 60 баллов		Цифра	Прописью	
1	2	3	4	5	7	8	9	10	11	12
1.	Иванов И.И.	6	8	6		19		4	хор.	
2.	Петров	7	7	6		10		4	удовл.	

	П.П.									
3.										

Структура оценивания домашних заданий

Критерии оценивания	Баллы
Аккуратность и полнота выполнения всех пунктов задания	0-1
Понимание логики выполнения задания и значения полученных результатов	0-1

Структура 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

Структура оценивания тестирования

Критерии оценивания	Баллы
Задание выполнено верно	0,5
Задание выполнено наполовину или вообще неверно	0

Структура оценивания на экзамене

Уровни оценивания	Критерии оценивания	Баллы
оценка «отлично»	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	41-60

Уровни оценивания	Критерии оценивания	Баллы
	оперировать ими; проявляя творческие 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.	
оценка «хорошо»	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	31-50

Уровни оценивания	Критерии оценивания	Баллы
	самостоятельную работу.	
оценка «удовлетворительно»	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.	21-40
оценка «неудовлетворительно»	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.	1-20