

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Тарасова Ирина Владимировна
Должность: Проректор по учебной работе
Дата подписания: 25.03.2022 16:36:23
Уникальный программный ключ:
8c45e14bf77dac42d4f8b124280a05e6949a00d3

**ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
ПРАВОСЛАВНЫЙ СВЯТО-ТИХОНОВСКИЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ
(ПСТГУ)**

*Факультет информатики и прикладной математики
Кафедра информатики*

УТВЕРЖДАЮ



Проректор по учебной работе

Тарасова И.В. /

_____ 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Практикум по олимпиадному программированию (часть 2)»

02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль подготовки:
Администрирование информационных систем

Квалификация выпускника: бакалавр

Форма обучения: очная

Москва, 2020 г.

Год начала обучения по учебному плану: 2019

1. Цели освоения дисциплины

Целью изучения дисциплины является формирование прочной теоретической базы для понимания нестандартных методов программирования, а также применения их на практике.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к блоку ФТД.02 образовательной программы.

Дисциплина изучается на 2 курсе, в 4 семестре.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины обучающийся должен продемонстрировать следующие результаты:

Коды компетенций	Наименование компетенции	Перечень планируемых результатов обучения по дисциплине
ПК-1	Способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий	В результате формирования компетенции обучающийся должен ЗНАТЬ: 1. отличия способов объявления объектов и классов в языках программирования Java и C++; 2. отличия способов перегрузки методов в языках программирования Java и C++; 3. способы работы с массивами объектов в Java; 4. способы реализации и назначение «static»-блоков в Java; 5. способы приведения типов в Java; 6. способы реализации абстрактных классов и интерфейсов в Java; 7. как использовать контейнерные классы в Java; 8. основные проблемы, возникающие при работе с необобщенными контейнерными классами в Java; 9. понятие обобщенного класса и способы решение проблемы необходимости явного приведения типов в Java; 10. понятия верхней и нижней границы формального типового параметра ограниченного обобщенного класса в Java; 11. понятие ковариантности массивов и инвариантности обобщенных классов в Java; 12. способы определения шаблонов обобщенных классов и их особенности в Java; 13. понятие полиморфизма на основе шаблонов обобщенных классов в Java; 14. способы реализации шаблонов, ограниченных

		<p>сверху и снизу в Java;</p> <ol style="list-style-type: none"> 15. способы перехвата нештатных ситуаций и иерархию классов исключений в Java; 16. понятие функтора в Java; 17. понятие монады в Java; 18. понятие пакетов в Java; 19. основы методов объектно-ориентированного программирования на примере языка программирования Java, а также понимать основные отличия от языка программирования C++; 20. понятия абстрактный класс, интерфейсы, вложенные классы, функциональные интерфейсы, обобщенные классы, шаблоны и обобщенные методы в языке Java; 21. классы исключений, перехват нештатных ситуаций в языке Java. <p>УМЕТЬ:</p> <ol style="list-style-type: none"> 1. реализовывать функциональные интерфейсы и лямбда-выражения; 2. реализовывать необобщенный класс на примере класса Stack; 3. реализовывать ограниченный обобщенный класс на примере класса Stack; 4. реализовывать перехват исключений; 5. реализовывать задачу поиска корней квадратного уравнения с использованием функтора; 6. реализовывать «заворачивание» и «разворачивание» Optional<T>; 7. реализовывать композицию частичных функций; 8. реализовывать потоки, как развитие идеи итераторов; 9. реализовывать интерфейсы для обеспечения возможности полиморфной обработки объектов класса на языке Java; 10. реализовывать обобщённые итераторы и экземплярные вложенные классы на языке Java; 11. использовать монады Optional и Stream в программах на языке Java; 12. разрабатывать программы на языке Java с графическим пользовательским интерфейсом на основе библиотеки swing.
--	--	---

		<p>ВЛАДЕТЬ:</p> <ol style="list-style-type: none"> 1. средой разработки Java Development Kit (JDK) версии 1.8 и выше в операционной системе Windows и Linux 2. интегрированной средой разработки программного обеспечения IntelliJ IDEA Community Edition версии не ниже 2021.3.1; 3. командным интерпретатором Bash в OS Linux и уметь работать на удаленном сервере с помощью программы PuTTY по протоколу SSH; 4. свободным пакетом построения 2d и 3d графиков GNUPlot; 5. средствами автоматизации выполнения процессов в Linux на базе интерпретатора Bash.
--	--	---

4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 2 зачетные единицы, 72 академических часа.

На занятия практического (семинарского) типа — 36 часов.

Самостоятельная работа составляет 36 часов.

5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1. Тематические разделы дисциплины и компетенции, которые формируются при их изучении

№ п/п	Наименование раздела дисциплины	Содержание раздела	Код формируемой компетенции
1.	Основы объектно-ориентированного программирования на языках Java и C++	Объекты и классы, поля, методы, экземплярные конструкторы, создание объектов, статические конструкторы, субтипизация, наследование, интерфейсы.	ПК-1
2.	Обобщения в Java и C++	Обобщенные классы, шаблоны, обобщенные методы.	ПК-1
3.	Исключения в Java	Классы исключений, перехват, порождение исключений.	ПК-1
4.	Категориальные классы в Java	Монады, монада Optional, монада Stream.	ПК-1

5.	Пакеты в Java	Создание пакета, импорт.	ПК-1
6.	Стороннее программное обеспечение	Автоматизация процессов в Linux, вспомогательное программное обеспечение, оболочки и среды.	ПК-1

5.2. Разделы дисциплины, виды учебных занятий и формы текущего контроля успеваемости

№ семестра	Наименование раздела дисциплины (модуля)	Трудоемкость в часах					Формы СРС	Формы текущего контроля	Формы текущего контроля с указанием баллов (при использовании балльной системы оценивания)
		Всего (вкл. СРС)	На контактную работу по видам учебных занятий		На СРС	Конт роль			
			Л	ПЗ					
1.	Основы объектно-ориентированного программирования на языках Java и C++	8		4	4			10	
2.	Обобщения в Java и C++	16		8	8			10	
3.	Исключения в Java	16		8	8			20	
4.	Категориальные классы в Java	16		8	8			10	
5.	Пакеты в Java	8		4	4			20	
6.	Стороннее программное обеспечение	8		4	4			10	
	Зачет						Зачет	20	
ИТОГО:		72		36	36			100	

Виды учебных занятий указаны в сокращенном виде: ЛР — Лабораторные работы, Л — лекция, ПЗ — практическое занятие (семинар), СРС — самостоятельная работа, Коллок. — Коллоквиумы, КР — контрольные работы.

6. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

Студентам выдается программа курса (примерный список вопросов к экзамену), список тем контрольных работ, список устных опросов и домашних заданий, критерии оценивания и способы повышения оценки, список литературы.

7. Проведение промежуточной аттестации обучающихся по дисциплине

7.1. Общие условия

Промежуточная аттестация по дисциплине – экзамен, проводится на основании результатов текущего контроля и результата, полученного на экзамене. Экзамен проводится в форме устного опроса.

Дисциплина оценивается по 100-балльной системе. Максимальное количество баллов, которое студент может набрать за один семестр – 80. Максимальное количество баллов, которое студент может набрать за ответ на экзамене – 20.

7.2. Критерии и шкалы оценивания результатов обучения по дисциплине

Код компетенции	Показатели достижения результатов обучения	Критерии и шкала оценивания			Перечень оценочных средств
		удовлетворительно	хорошо	Отлично	
ПК-1	<p>В результате формирования компетенции обучающийся должен</p> <p>ЗНАТЬ:</p> <ol style="list-style-type: none"> 1. отличия способов объявления объектов и классов в языках программирования Java и C++; 2. отличия способов перегрузки методов в языках программирования Java и C++; 3. способы работы с массивами объектов в Java; 4. способы реализации и назначение «static»-блоков в Java; 5. способы приведения типов в Java; 6. способы реализации абстрактных классов и интерфейсов в Java; 7. как использовать контейнерные классы в Java; 8. основные проблемы, возникающие при работе с необобщенными контейнерными классами в Java; 9. понятие обобщенного класса и способы решение проблемы необходимости явного приведения типов в Java; 10. понятия верхней и нижней границы формального типового параметра ограниченного обобщенного класса в Java; 11. понятие ковариантности массивов и инвариантности обобщенных классов в Java; 12. способы определения шаблонов обобщенных классов и их особенности в Java; 	Удовлетворительное владение основными понятиями Умение применять знания в стандартной ситуации	хорошее владение основными понятиями Умение применять знания в сложной стандартной ситуации	свободное владение основными понятиями Умение применять знания в сложной нестандартной ситуации	Зачет

	<p>13. понятие полиморфизма на основе шаблонов обобщенных классов в Java;</p> <p>14. способы реализации шаблонов, ограниченных сверху и снизу в Java;</p> <p>15. способы перехвата нештатных ситуаций и иерархию классов исключений в Java.</p> <p>16. понятие функтора в Java;</p> <p>17. понятие монады в Java;</p> <p>18. понятие пакетов в Java;</p> <p>19. основы методов объектно-ориентированного программирования на примере языка программирования Java, а также понимать основные отличия от языка программирования C++;</p> <p>20. понятия абстрактный класс, интерфейсы, вложенные классы, функциональные интерфейсы, обобщенные классы, шаблоны и обобщенные методы в языке Java;;</p> <p>21. классы исключений, перехват нештатных ситуаций в языке Java.</p> <p>УМЕТЬ:</p> <ol style="list-style-type: none"> 1. реализовывать функциональные интерфейсы и лямбда-выражения; 2. реализовывать необобщенный класс на примере класса Stack; 3. реализовывать ограниченный обобщенный класс на примере класса Stack; 4. реализовывать перехват исключений; 5. реализовывать задачу поиска корней квадратного уравнения с использованием функтора; 6. реализовывать «заворачивание» и «разворачивание» Optional<T>; 7. реализовывать композицию частичных функций; 8. реализовывать потоки, как развитие идеи итераторов; 9. реализовывать интерфейсы для обеспечения возможности полиморфной обработки объектов класса на языке Java; 10. реализовывать обобщенные итераторы и экземплярные вложенные классы на языке Java; 				
--	--	--	--	--	--

	<p>11. использовать монады Optional и Stream в программах на языке Java;</p> <p>12. разрабатывать программы на языке Java с графическим пользовательским интерфейсом на основе библиотеки swing.</p> <p>ВЛАДЕТЬ:</p> <ol style="list-style-type: none"> 1. средой разработки Java Development Kit (JDK) версии 1.8 и выше в операционной системе Windows и Linux 2. интегрированной средой разработки программного обеспечения IntelliJ IDEA Community Edition версии не ниже 2021.3.1; 3. командным интерпретатором Bash в OS Linux и уметь работать на удаленном сервере с помощью программы PuTTY по протоколу SSH; 4. свободным пакетом построения 2d и 3d графиков GNUPlot; 5. средствами автоматизации выполнения процессов в Linux на базе интерпретатора Bash. 				
--	--	--	--	--	--

7.3. Оценочные средства для промежуточной аттестации

Промежуточная аттестация (в конце семестра)

Проводится на основании результатов текущей аттестации и результата, полученного на экзамене. Экзамен проходит в форме устного опроса.

Примерный перечень вопросов к зачету:

1. JIT-компиляция. Переносимость Java-программ.
2. Установка JDK на компьютер под управлением ОС Linux.
3. Компиляция и запуск Java-программ.
4. Понятие объекта. Характеристика объекта, позволяющая организовать позднее связывание при вызове методов.
5. Понятие инкапсуляции. Модификаторы доступа в языке Java.
6. Понятие класса. Синтаксис объявления класса в Java. Категории членов класса.
7. Операция доступа к членам класса в Java. Обращение к полям и вызов методов.
8. Понятие экземплярного поля. Объявление экземплярных полей в Java.
9. Понятие статического поля. Объявление статических полей в Java.
10. Понятие экземплярного метода. Объявление и вызов экземплярных методов в Java.
11. Понятие статического метода. Объявление и вызов статических методов в Java.

12. Понятие сигнатуры метода. Перегрузка методов.
13. Понятия раннего и позднего связывания. Виртуальные методы.
14. Понятие экземплярного конструктора. Объявление экземплярных конструкторов в Java.
15. Конструктор по умолчанию.
16. Размещение объектов Java в памяти. Объектные ссылки.
17. Создание объектов и массивов в Java. Массивовые литералы.
18. Понятие статического конструктора, «static»-блоки в Java.
19. Понятие подтипа. Явная и неявная субтипизация.
20. Понятие наследования. Синтаксис наследования в Java. Вызов конструктора базового класса из конструктора производного класса.
21. Операция динамического приведения типа в Java.
22. Понятие переопределения метода. Переопределение методов в Java.
23. Понятия абстрактного метода и абстрактного класса. Их объявления в Java.
24. Понятие интерфейса. Объявление интерфейсов в Java. Наследование интерфейсов. Реализация интерфейса.
25. Понятия экземплярных и статических вложенных классов. Их объявления в Java.
26. Понятие обобщённого класса. Объявление обобщённых классов в Java.
27. Понятие ограниченного обобщённого класса. Задание верхней границы типового параметра в Java.
28. Ковариантность массивов в Java.
29. Инвариантность обобщённых классов в Java.
30. Понятие шаблона обобщённого класса в Java. Частичные шаблоны.
31. Шаблоны обобщённых классов в Java, ограниченные сверху.
32. Шаблоны обобщённых классов в Java, ограниченные снизу.
33. Объявление и вызов обобщённых методов в Java.
34. Понятия нештатной, исключительной и ошибочной ситуаций. Два способа перехвата нештатных ситуаций.
35. Понятие исключения. Иерархия классов исключений в Java.
36. Операторы перехвата исключений в Java.
37. Порождение исключения в Java. Модификатор «throws» в объявлении метода.
38. Понятие пакета в Java. Имена пакетов и квалифицированные имена классов. Создание пакетов. Пакет по умолчанию.
39. Правила видимости для классов пакетов Java и их членов. Импорт классов. Импорт статических членов класса.
40. Объявление локальных и анонимных классов в Java. Лямбда-выражения.

41. Ссылочный тип данных в C++. Константные ссылки.
42. Объявление класса в C++. Секции в объявлении класса.
43. Объявления экземплярных и статических полей в C++. Определения статических полей.
44. Объявления методов в C++. Виртуальные и абстрактные методы.
45. Объявление экземплярных конструкторов в C++.
46. Создание объектов в динамической памяти в C++. Удаление объектов.
47. Создание массивов в динамической памяти в C++. Удаление массивов.
48. Объявление деструкторов в C++. Как осуществляется вызов деструктора.
49. Объекты C++ в автоматической и глобальной памяти. Автоматический вызов деструкторов.
50. Объекты C++ в полях других объектов, их инициализация.
51. Проблема копирования объектов. Конструктор копий и перегрузка операции присваивания.
52. Одиночное наследование в C++, вызов конструктора базового класса.
53. Переопределение методов в C++.
54. Динамическое приведение типов в C++.
55. Множественное наследование в C++, вызов конструкторов базовых классов.
56. Разрешение противоречий в именах наследуемых членов классов при множественном наследовании в C++.
57. Понятие иерархии наследования. Классы противоречия. Основная проблема противоречивых иерархий.
58. Виртуальное наследование в C++.
59. Понятие шаблона в C++. Виды параметров шаблона. Значения параметров шаблона по умолчанию.
60. Инстанциация шаблона в C++. Выведение фактических параметров шаблона при инстанциации. Требования к фактическим параметрам шаблона.
61. Специализация шаблона функции в C++.

7.4. Шкала перевода оценок

Оценка за ответ на зачете выставляется на основе следующих критериев:

Шкалы оценивания	Критерии
Баллы	
18-20	<p>Обучающийся</p> <ul style="list-style-type: none"> – полно излагает изученный материал, дает правильное определение понятий; – обнаруживает понимание материала, может обосновать свои суждения, привести необходимые примеры не только из учебных пособий, но и самостоятельно составленные; – излагает материал последовательно и правильно с точки зрения норм литературного языка.

15-17	Обучающийся – полно излагает изученный материал, дает правильное определение понятий; – обнаруживает понимание материала, может обосновать свои суждения, привести необходимые примеры; – допускает 1-2 ошибки и 1-2 недочета в последовательности и излагаемого.
12-14	Обучающийся – обнаруживает знание и понимание основных положений; – но излагает материал неполно и допускает неточности в определении понятий или формулировках; – не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры
0	Обучающийся: – обнаруживает незнание большей части соответствующего раздела изучаемого материала, допускает ошибки в формулировке определений, искажающие их смысл, беспорядочно; – неуверенно излагает материал.

Итоговая оценка (с учетом зачета и результатов текущего контроля) выставляется на основе следующих критериев (по суммарному количеству баллов за зачет и текущий контроль):

Форма промежуточной аттестации	Шкала оценивания		Критерии оценивания
	Оценка	В баллах по 100-балльной шкале	
Зачет	Зачтено	Не менее 61	Сданы все лабораторные работы, набрано (*) не менее 61 балла
Зачет	Не зачтено	Менее 61	Сданы не все лабораторные работы или набрано (*) менее 61 балла

(*) набранные баллы считаются в сумме за текущий контроль и ответ на зачете.

8. Перечень образовательных технологий

В процессе преподавания дисциплины используются следующие образовательные технологии:

1. Лекции с обсуждением проблемных мест,
2. Практические занятия с решением задач,
3. Разбор домашних заданий с элементами дискуссии и взаимопомощи обучающихся друг другу,
4. Устные опросы.

9. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) Основная литература

1. Г. Шилдт. Java 8. Руководство для начинающих. 6-е издание. – М.: Издательский дом

«Вильямс», 2015.

2. Б. Страуструп. Язык программирования C++. Специальное издание. – М.: Бином, 2011.

б) Дополнительная литература

1. Дж. Гослинг и др. Язык программирования Java SE 8. Подробное описание. 5-е издание. – М.: Издательский дом «Вильямс», 2015.
2. Б. Эккель. Философия Java. Библиотека программиста. 4-е издание. – СПб.: Питер, 2009.
3. Б. Страуструп. Дизайн и эволюция C++. – М.: ДМК Пресс; СПб.: Питер, 2006.

10. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины

Подключение к интернет для поиска ресурсов, имеющихся в свободном доступе.

11. Методические указания для обучающихся по освоению дисциплины

Студентам выдается программа курса (примерный список вопросов к экзамену), список тем контрольных работ, устных опросов и домашних заданий. Критерии оценивания и способы повышения оценки. Список литературы.

12. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

OS Windows 7 и выше или Linux дистрибутив Ubuntu версии не ниже 20.04, Java Development Kit (JDK) версии 1.8 и выше, IntelliJ IDEA Community Edition версии не ниже 2021.3.1

13. Описание материально-технической базы, необходимой для осуществления Компьютерный класс, оснащенный рабочими станциями

Компьютерный класс, оснащенный типовыми рабочими станциями Pentium 4+, 2,4 GHz, 250 GB HD, OS Windows 7 и выше или Linux дистрибутив Ubuntu версии не ниже 20.04. Персональный компьютер на каждого студента.

Разработчики программы:

доц. к.ф.-м.н. Посевин Д. П.

Рецензент:

профессор, к.т.н. Соловьев В.П.

Программа одобрена на заседании кафедры Информатики от «29» мая 2020 года, протокол № 05-20