МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Институт вычислительной математики и информационных технологий

УТВЕРЖДАЮ
Проректор по образовательной деятельности КФУ
Проф. Д.А. Таюрский
_____ «___»_____20__г.

подписано электронно-цифровой подписью

Программа дисциплины

Технологии программирования Б1.О.03

Направление подготовки: 01.04.04 - Прикладная математика
Профиль подготовки: Классические и квантовые методы обработки информации
Квалификация выпускника: магистр
Форма обучения: очное
Язык обучения: русский
Год начала обучения по образовательной программе: 2019
Автор(ы): Аблаев Ф.М.
Рецензент(ы): Байрашева В.Р.

СОГЛАСОВАНО:
Заведующий(ая) кафедрой: Аблаев Ф. М.
Протокол заседания кафедры No ___ от "____" _____ 20__г.
Учебно-методическая комиссия Института вычислительной математики и информационных технологий:
Протокол заседания УМК No _____ от "____" _____ 20__г.

Содержание

Программу дисциплины разработал(а)(и) заведующий кафедрой, д.н. (профессор) Аблаев Ф.М. (кафедра теоретической кибернетики, отделение фундаментальной информатики и информационных технологий), Farid.Ablayev@kpfu.ru

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Выпускник, освоивший дисциплину, должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-1	Способен обобщать и критически оценивать опыт и результаты научных исследований в области прикладной математики
ОПК-2	Способен разрабатывать и развивать математические методы моделирования объектов, процессов и систем в области профессиональной деятельности
ОПК-3	Способен разрабатывать наукоемкое программное обеспечение для автоматизации систем и процессов, а также развивать информационно-коммуникационные технологии
УК-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки

Выпускник, освоивший дисциплину:

Должен знать:

специфику программирования на языке Java, принципы построения языка , его особенности в сравнении с другими языками, основные этапы и тенденции развития ООП на языке Java, возможности, реализуемые технологией Java в интернете, различные паттерны проектирования программного обеспечения, реализованные в стандартных библиотеках Java, приемы контроля входных данных приложения.

Должен уметь:

ориентироваться в технологии JDBC, реализации компонентов JFC1, использовать компоненты других стандартных библиотек Java для решения профессиональных задач,

создавать иерархию классов приложения, создавать диаграммы UML, извлекать полезную научно-техническую информацию из электронных библиотек, реферативных журналов, сети Интернет.

Должен владеть:

теоретическими знаниями об основных компонентах языка и их использовании при написании программ, навыками самостоятельной работы при разработке и отладке программ, навыками работы в средах разработки программного обеспечения NetBeans/Eclipse и IntelliJ IDEA для решения профессиональных задач.

Должен демонстрировать способность и готовность:

приобрести навыки применения ООП концепций и разработки приложений, используя ключевые основные сервисы языка

2. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования

Данная учебная дисциплина включена в раздел "Б1.О.03 Дисциплины (модули)" основной профессиональной образовательной программы 01.04.04 "Прикладная математика (Классические и квантовые методы обработки информации)" и относится к обязательным дисциплинам.
Осваивается на 1 курсе в 2 семестре.

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

Общая трудоемкость дисциплины составляет 8 зачетных(ые) единиц(ы) на 288 часа(ов).

Контактная работа - 36 часа(ов), в том числе лекции - 0 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 36 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 216 часа(ов).

Контроль (зачёт / экзамен) - 36 часа(ов).

Форма промежуточного контроля дисциплины: экзамен во 2 семестре.

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

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Введение. Особенности языка Java	2	0	0	2	11
2.	Тема 2. Конструкции языка. Примитивные и ссылочные типы данных.	2	0	0	4	15
3.	Тема 3. ООП в Java. Пакет java.lang.	2	0	0	2	15
4.	Тема 4. Коллекции.	2	0	0	2	15
5.	Тема 5. Исключения.Модульное тестирование.	2	0	0	4	15
6.	Тема 6. Ввод и вывод.	2	0	0	4	15
7.	Тема 7. Java Foundation Classes (JFC)	2	0	0	4	15
8.	Тема 8. Функциональное программирование. Лямбда-выражения.	2	0	0	2	15
9.	Тема 9. Многопоточные приложения	2	0	0	2	15
10.	Тема 10. Работа с базами данных средствами JDBC.	2	0	0	2	15
11.	Тема 11. Система сборки Maven.	2	0	0	2	15
12.	Тема 12. Возможности Stream API и их использование	2	0	0	2	15
13.	Тема 13. Фреймворк Spring.	2	0	0	4	40
	Итого		0	0	36	216

4.2 Содержание дисциплины

Тема 1. Введение. Особенности языка Java

Создание языка Java. Особенности языка. Сборка мусора. Безопасность. Многопоточность. Редакции языка Java. их возможности. Сторонние библиотеки. Системы сборки. Платформонезависимость. Среды разработки. Альтернативные языки, использующие JVM. Инструментарий JDK, компилятор, отладчик, дизассемблер и пр.

Тема 2. Конструкции языка. Примитивные и ссылочные типы данных.

Структура языка. Основные конструкции, операции, операторы. Примитивные типы данных. Преобразование типов: явное и неявное. Автоматическое расширение. Классы BigInteger и BigDecimal. Ссылочные типы данных: массивы (объявление, создание, инициализация, многомерные массивы). перечисления. Класс java.util.Array.

Тема 3. ООП в Java. Пакет java.lang.

Основы ООП. Объекты и классы, конструкторы, инкапсуляция, наследование, полиморфизм. Правила оформления: объявление класса, модификаторы доступа. Передача параметров. Внутренние классы. Абстрактные классы. Интерфейсы и их роль в обеспечении полиморфизма. Реализация парадигмы множественного наследия.

Пакет java.lang.

Тема 4. Коллекции.

Параметризованные методы и классы. Использование коллекций, иерархия интерфейсов коллекций, виды коллекций. Интерфейс Collection. Интерфейс List. Классы java.util.ArrayList и java.util.LinkedList. Интерфейсы Iterator и ListIterator. Интерфейс Set. Классы HashSet, TreeSet и LinkedHashSet. Интерфейс Map.

Тема 5. Исключения.Модульное тестирование.

Подходы к обработке ошибок. Проверяемые и непроверяемые исключения, обработка исключений, иерархия исключений. Описание исключений. Класс Throwable. Генерация исключений, объявление контролируемых исключений. Перехват исключений. Обоснование модульного тестирования и рекомендации к написанию тестов.

Тема 6. Ввод и вывод.

Чтение входных данных. Форматирование выходных данных. Файловый ввод/вывод. Иерархии классов ввода/вывода. Символьный и байтовый ввод данных. Классы - Reader, Writer, InputStream и - OutputStream. Базовые методы чтения и записи. Файлы с произвольным доступом. Блокирование файлов. Сериализация. Особенности Java NIO.

Тема 7. Java Foundation Classes (JFC)

Компоненты JFC: AVT, Java2D, Swing и т.д. Основные свойства Swing: компоненты, контейнеры, панели. Основные компоненты (Buttons, Labels, Text fields, Text areas, Check boxes, Radio buttons, Drop-down lists, List boxes, Tabbed panes, Menus, Message Boxes, Dialog Boxes). Обработка основных событий. Иконки и изображения. Layers, Panels, использование Layout Managers. Модель обработки событий. Создание окон, создание меню. Swing компоненты JTree, JTable, JSlider, JProgressBar.

Тема 8. Функциональное программирование. Лямбда-выражения.

Функциональные интерфейсы и их синтаксис. Типы ссылок. Присвоение лямбда-выражения. Синтаксис лямбда-выражения. Ключевые моменты понятия лямбды. Примеры использования лямбда-выражения. Зачем нужны лямбды. Ссылки на методы. Ссылки на конструктор. Область действия переменной. Захват значений в лямбда-выражении.

Тема 9. Многопоточные приложения

Многопоточность в Java. Понятие потока, исполнители, получение значений из потоков, потоки-демоны, присоединение к потоку, взаимодействие потоков.

Класс Thread и интерфейс Runnable. Создание потока. Завершение потока. Приоритеты потоков. Синхронизация. Взаимодействие потоков. Методы notify(), wait(), notifyAll()

java.util.concurrent ? библиотека для многопоточного программирования. Concurrent Collections. Callable и Future.

Тема 10. Работа с базами данных средствами JDBC.

JDBC и его архитектура, конфигурирование. Подключение драйверов. Важнейшие интерфейсы и их функциональность: DriverManager, Driver, Connection, Statement, Metadata, ResultSet. Структура стандартной программы обработки БД. Выполнение операторов SQL, анализ исключений, транзакции. java.sql.PreparedStatement и java.sql.CallableStatement. ORM и Hibernate (библиотека).

Тема 11. Система сборки Maven.

Основные инструменты для сборки на платформе Java. Структура проекта Maven. Основные понятия Maven: POM (Project Object Model), зависимости, Plagins (плагины), Artefact (артефакт), Repository (репозиторий), Coordinates (координаты), Archetype (архетип). Жизненный цикл сборки: фазы сборки. Основные преимущества Maven.

Тема 12. Возможности Stream API и их использование

Способы создание стримов. Классический способ: Создание стрима из коллекции, создание стрима из значений, создание стрима из массива, создание стрима из файла (каждая строка в файле будет отдельным элементом в стриме, создание стрима из строки), с помощью Stream.builder, создание параллельного стрима, создание бесконечных стрима с помощью Stream.iterate, создание бесконечных стрима с помощью Stream.generate . Методы работы со стримами: конвейерный и терминальный. Примеры использования различных стримов

Тема 13. Фреймворк Spring.

Что такое framework Spring. Модули Spring. Внедрение зависимостей, модули Beans и Core. Использование контейнера. Именование бинов и старт контекста. Внутренние бины. Инициализация и удаление. Возможности контекста. Именование бинов и старт контекста. Внедрение через property. Внедрение коллекций. Конфигурация с помощью аннотаций.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301).

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений".

Положение от 29 декабря 2018 г. № 0.1.1.67-08/328 "О порядке проведения текущего контроля успеваемости и промежуточной аттестации обучающихся федерального государственного автономного образовательного учреждения высшего образования "Казанский (Приволжский) федеральный университет".

Положение № 0.1.1.67-06/241/15 от 14 декабря 2015 г. "О формировании фонда оценочных средств для проведения текущей, промежуточной и итоговой аттестации обучающихся федерального государственного автономного образовательного учреждения высшего образования "Казанский (Приволжский) федеральный университет"".

Положение № 0.1.1.56-06/54/11 от 26 октября 2011 г. "Об электронных образовательных ресурсах федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"".

Регламент № 0.1.1.67-06/66/16 от 30 марта 2016 г. "Разработки, регистрации, подготовки к использованию в учебном процессе и удаления электронных образовательных ресурсов в системе электронного обучения федерального государственного автономного образовательного учреждения высшего образования "Казанский (Приволжский) федеральный университет"".

Регламент № 0.1.1.67-06/11/16 от 25 января 2016 г. "О балльно-рейтинговой системе оценки знаний обучающихся в федеральном государственном автономном образовательном учреждении высшего образования "Казанский (Приволжский) федеральный университет"".

Регламент № 0.1.1.67-06/91/13 от 21 июня 2013 г. "О порядке разработки и выпуска учебных изданий в федеральном государственном автономном образовательном учреждении высшего профессионального образования "Казанский (Приволжский) федеральный университет"".

6. Фонд оценочных средств по дисциплине (модулю)

6.1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения

Этап	Форма контроля	Оцениваемые компетенции	Темы (разделы) дисциплины
Семестр 2			
	Текущий контроль		

Этап	Форма контроля	Оцениваемые компетенции	Темы (разделы) дисциплины
1	Компьютерная программа	ОПК-1 , ОПК-2	2. Конструкции языка. Примитивные и ссылочные типы данных. 4. Коллекции. 5. Исключения.Модульное тестирование. 6. Ввод и вывод.
2	Лабораторные работы	УК-1 , ОПК-3	7. Java Foundation Classes (JFC) 8. Функциональное программирование. Лямбда-выражения. 9. Многопоточные приложения 10. Работа с базами данных средствами JDBC.
3	Тестирование	ОПК-2 , УК-6	11. Система сборки Maven. 12. Возможности Stream API и их использование 13. Фреймворк Spring.
	Экзамен	ОПК-1, ОПК-2, ОПК-3, УК-1, УК-6	

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

Форма контроля	Критерии оценивания				Этап
	Отлично	**Хорошо**	**Удовл.**	**Неуд.**	
Семестр 2					
Текущий контроль					
Компьютерная программа	Высокий уровень умений и навыков программирования, в том числе моделирования, алгоритмизации, использования языка программирования. Поставленная задача полностью решена.	Хороший уровень умений и навыков программирования, в том числе моделирования, алгоритмизации, использования языка программирования. Поставленная задача в основном решена.	Удовлетворительный уровень умений и навыков программирования, в том числе моделирования, алгоритмизации, использования языка программирования. Поставленная задача решена частично.	Недостаточный уровень умений и навыков программирования, в том числе моделирования, алгоритмизации, использования языка программирования. Поставленная задача не решена.	1
Лабораторные работы	Оборудование и методы использованы правильно. Проявлена превосходная теоретическая подготовка. Необходимые навыки и умения полностью освоены. Результат лабораторной работы полностью соответствует её целям.	Оборудование и методы использованы в основном правильно. Проявлена хорошая теоретическая подготовка. Необходимые навыки и умения в основном освоены. Результат лабораторной работы в основном соответствует её целям.	Оборудование и методы частично использованы правильно. Проявлена удовлетворительная теоретическая подготовка. Необходимые навыки и умения частично освоены. Результат лабораторной работы частично соответствует её целям.	Оборудование и методы использованы неправильно. Проявлена неудовлетворительная теоретическая подготовка. Необходимые навыки и умения не освоены. Результат лабораторной работы не соответствует её целям.	2
Тестирование	86% правильных ответов и более.	От 71% до 85 % правильных ответов.	От 56% до 70% правильных ответов.	55% правильных ответов и менее.	3

Форма контроля	Критерии оценивания				Этап
	Отлично	Хорошо	Удовл.	Неуд.	
Экзамен	Обучающийся обнаружил всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой дисциплины, усвоил взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявил творческие способности в понимании, изложении и использовании учебно-программного материала.	Обучающийся обнаружил полное знание учебно-программного материала, успешно выполнил предусмотренные программой задания, усвоил основную литературу, рекомендованную программой дисциплины, показал систематический характер знаний по дисциплине и способен к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.	Обучающийся обнаружил знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справился с выполнением заданий, предусмотренных программой, знаком с основной литературой, рекомендованной программой дисциплины, допустил погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладает необходимыми знаниями для их устранения под руководством преподавателя.	Обучающийся обнаружил значительные пробелы в знаниях основного учебно-программного материала, допустил принципиальные ошибки в выполнении предусмотренных программой заданий и не способен продолжить обучение или приступить по окончании университета к профессиональной деятельности без дополнительных занятий по соответствующей дисциплине.	

6.3 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

Семестр 2

Текущий контроль

1. Компьютерная программа

Темы 2, 4, 5, 6

1. Ввести n строк с консоли, найти самую короткую строку. Вывести эту строку и ее длину.
2. Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания их длин, а также (второй приоритет) значений этих их длин.
3. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше средней, также их длины.
4. В каждом слове текста k-ю букву заменить заданным символом. Если k больше длины слова, корректировку не выполнять.
5. В русском тексте каждую букву заменить ее номером в алфавите. В одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.
6. Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.
7. Из текста удалить все слова заданной длины, начинающиеся на согласную букву.
8. В тексте найти все пары слов, из которых одно является об?ращением другого.
9. Найти и напечатать, сколько раз повторяется в тексте каждое слово.
10. Найти, каких букв, гласных или согласных, больше в каждом предложении текста.
11. Выбрать три разные точки заданного на плоскости множества точек, составляющие треугольник наибольшего периметра.
12. Найти такую точку заданного на плоскости множества точек, сумма расстояний от которой до остальных минимальна.
13. Выпуклый многоугольник задан на плоскости перечислением координат вершин в порядке обхода его границы. Определить площадь многоугольника.
14. Ввести строки из файла, записать их в стек. Вывести строки в файл в обратном порядке.
15. Ввести число, занести его цифры в стек. Вывести в число, у которого цифры идут в обратном порядке.
16. Сложить два многочлена заданной степени, если коэффициенты многочленов хранятся в объекте HashMap.

17. Создать стек из элементов каталога.
18. Не используя вспомогательных объектов, переставить отрицательные элементы данного списка в конец, а положительные - в начало этого списка.
19. Организовать вычисления в виде стека.
20. Выполнить попарное суммирование произвольного конечного ряда чисел следующим образом: на первом этапе суммируются попарно рядом стоящие числа, на втором этапе суммируются результаты первого этапа и т.д. до тех пор, пока не останется одно число.
21. Задать два стека, поменять информацию местами.
22. Определить класс Stack. Объявить объект класса. Ввести последовательность символов и вывести ее в обратном порядке.
23. Умножить два многочлена заданной степени, если коэффициенты многочленов хранятся в списках.
24. Определить класс Set на основе множества целых чисел, n = размер. Создать методы для определения пересечения и объединения множеств.

2. Лабораторные работы

Темы 7, 8, 9, 10

Задание №1. Контейнеры, IO потоки, классы Object и String.

Написать программу, которая будет принимать в качестве аргумента имя текстового файла, и выводить файл с колонками:

1. Слово.
2. Частота.
3. Частота (в %).

Файл должен быть упорядочен по убыванию частоты, то есть самые частые слова должны идти в начале. Разделителями считать все символы кроме букв и цифр.

Методические указания:

- Использовать класс java.lang.StringBuilder для построения слов.
- Для чтения из файла удобно использовать: java.io.InputStreamReader, например:

Reader reader = null; try { reader = new InputStreamReader(new FileInputStream("FILE NAME")); //read the data here } catch (IOException e) { System.err.println("Error while reading file: " + e.getLocalizedMessage()); } finally { if (null != reader) { try { reader.close(); } catch (IOException e) { e.printStackTrace(System.err); } } }

- Для определения класса символа использовать метод Character.isLetterOrDigit.
- Для хранения статистики в памяти можно использовать одну из реализаций интерфейса java.util.Set, который должен будет хранить объекты специального класса. Данный класс должен содержать слово и счѐтчик. В случае использования java.util.HashSet класс также должен реализовать методы equals, hashCode.

Теоретические сведения: Контейнеры стандартной библиотеки расположены в пакете java.util. IO классы (потоки ввода-вывода) располагаются в пакете java.io.

Основные интерфейсы:

1. Set ? множество без дубликатов и без доступа по индексу.
2. Map ? множество пар ключ-значение, где ключи не повторяются.

Их основные реализации:

1. HashMap, HashSet ? реализации на основе функции hashCode.
2. TreeMap, TreeSet ? реализация на основе бинарного дерева.

Ключи (элементы) должны реализовывать интерфейс Comporable, иначе необходимо передавать в контейнер при его создании объект, реализующий интерфейс Comparator. Хранимые в данных контейнерах данные упорядочены. Лучшее время поиска, но большее накладные расходы на вставку, чем на основе функции hashCode.

Задание №2. Шаблон проектирования ?MVC?, графический интерфейс пользователя (GUI).

Вариант №1 (Тетрис).

Постановка задачи Написать аналог игры ?Тетрис? (—Tetris‖). Набор фигур стандартный ? все возможные вариации связных многоугольников, составленных из 4-х квадратов. Архитектура программы должна быть основана на паттерне MVC (Mode-View-Controller).

Требования к программе

1. Игра должна поддерживать таблицу рекордов.
2. Пользователю должны быть доступны команды: Exit, About, New Game, High Scores.

Вариант №2 (Сапёр).

Постановка задачи Написать аналог игры ?Сапер? (—Minesweeper‖) из состава стандартных программ для Windows OS. Архитектура программы должна быть основана на паттерне MVC (ModeView-Controller). Программа должна иметь два интерфейса ? текстовый и графический, причем оба интерфейса должны использовать одну и ту же игровую модель. Т.е. классы данных и логики должны быть одинаковые для текстового и графического интерфейсов.

Требования к программе:

1. Размер поля и количество мин можно изменить. По умолчанию поле размером 9x9 и количество мин 10.
2. Игра должна поддерживать таблицу рекордов.

3. Пользователю должны быть доступны команды: Exit, About, New Game, High Scores.
4. Отчет времени должен быть реализован отдельным потоком.

Реализация текстового UI

1. Команды пользователя вводятся с консоли, ячейки нумеруются от ноля
2. После каждого хода игрока все игровое поле распечатывается на экран целиком

Реализация графического UI

1. Мины и флажки отображать с помощью картинок.
2. При формировании окна игры использовать класс LayoutManager.

Для расположения элементов на игровой панели рекомендуется использовать класс GridBagLayout. Для расположения ячеек поля рекомендуется использовать класс GridLayout.

Методические указания:

- Для реализации пользовательского интерфейса использовать библиотеку Swing (классы из пакета javax.swing.*).
- Работа с компонентами пользовательского интерфейса (классами библиотеки Swing) должна проходить только из UI потока.
- Для отображения диалоговых окон рекомендуется использовать класс JOptionPane.

Задание №3. Многопоточное программирование. ?Эмулятор работы фабрики?

1. Постановка задачи

Напишите приложение, эмулирующее работу фабрики по сборке автомашин. Машина состоит из 3-х частей: кузов, двигатель и аксессуары. Машину надо собрать и отвезти на склад, откуда она поступает дилерам. Процесс работы фабрики показан на картинке:

Рисунок 1 Схема работы фабрики по производству автомашин (файл Рисунок.pdf)

2. Требования к программе

1. Все склады имеют определенный размер, который нельзя превышать. Размеры складов, количество сборщиков, поставщиков и дилеров задаются в конфигурационном файле. Приложение предоставляет графический интерфейс (библиотека Swing), где можно смотреть основные параметры работы фабрики и контролировать процесс.
2. Каждый сборщик, поставщик и дилер должен работать в отдельном потоке. Для синхронизации и ожидания событий необходимо использовать мониторы синхронизации (notify(), notifyAll(), wait()). Наличие процедуры ожидания в виде цикла автоматически ведет к непринятию задания. Каждая деталь - это отдельный объект. Хранить просто количество изделий/деталей нельзя - необходимо хранить непосредственно объекты. Каждый объект должен иметь уникальный идентификатор для отслеживания.
3. Потоки, которые представляют поставщиков деталей, поставляют одну деталь раз в N миллисекунд. Если какой-то склад деталей полон, то поставщик ожидает освобождения места для деталей (используя методы wait(), notify()). Скорость работы поставщиков определяется 3-мя ползунками (для каждого типа деталей). Должно отображаться кол-во деталей на каждом из складов в текущий момент и кол-во деталей, произведенных поставщиками (для поставщиков аксессуаров можно общий показывать).
4. Потоки, которые представляют дилеров, запрашивают со склада готовой продукции 1 машину в M миллисекунд. Скорость запрашивания машин можно регулировать ползунком в интерфейсе окна. Интерфейс также должен показывать кол-во произведенных машин (вообще) и кол-во машин на складе в данный момент. При отправке машины дилеру информация о покупке должна писаться в лог работы фабрики (в файл) в виде строки:

<Time>: Dealer <Number>: Auto <ID> (Body: <ID>, Motor: <ID>, Accessory: <ID>)

Включение/отключение лога контролируется с помощью специального параметра в конфигурационном файле.

5. Поток контроллера склада готовой продукции просыпается при любом отправке машины со склада продукции. Он анализирует состояние склада и передает запрос на изготовление новых машин (в случае необходимости) на фабрику.
6. На фабрике работает несколько потоков (сборщиков) в рамках ThreadPool. Задачами для ThreadPool являются запросы на создание новых машин (от контроллера склада готовых изделий). При выполнении такой задачи сборщик должен взять по одной детали, необходимой для сборки машины, с соответствующих складов. Если на складе нет нужной детали, то поток ждет поставки. Собирая новую машину, рабочий создает новый объект и с помощью всех необходимых объектов, представляющих детали. После этого объект отправляется на склад готовой продукции. Если склад полон, то рабочий ждет освобождения места для новой машины. Интерфейс должен отображать, сколько всего было сделано машин и сколько задач еще ждут исполнителя (в очереди задач ThreadPool).
7. Конфигурационный файл должен предоставлять настройки для задания вместимости всех складов и количестве всех типов потоков. Примерный список параметров в конфигурационном файле (просьба использовать свои имена):

StorageBodySize=100
StorageMotorSize=100
StorageAccessorySize=100
StorageAutoSize=100
AccessorySuppliers=5
Workers=10
Dealers=20

LogSale=true
3. Особенности реализации
1. Классы объектной модели (деталь, склад, поставщик, сборщик и т.д.) не должны зависеть от библиотеки графического интерфейса SWING.
2. Пул потоков (ThreadPool) нужно реализовать в отдельном пакете. ThreadPool должен выполнять абстрактные задачи и не зависеть от реализации (задача на сборку автомашины в фабрике).
3. Программа должна корректно завершаться. При получении сигнала о закрытии окна все потоки должны корректно прерываться, и лог-файл закрываться.

3. Тестирование

Темы 11, 12, 13
Вопрос 1. Каков будет результат выполнения программы

```
1. public class Test {
2. public Test() {
3. }
4.
5. public static void main(String[] args) {
6. Test test = new Test();
7. int i = 5;
8. while(i = 5){
9. System.out.println(i++);
10. }
11. }
12. }
```

Варианты ответов
1. а) Компилятор выдаст сообщение об ошибке в строке 8
2. б) На консоль будут последовательно выведены значения 01234
3. в) На консоль будут последовательно выведены значения 43210
4. г) Программа откомпилируется, но на консоль ничего выведено не будет

Вопрос 2. Приведенная ниже программа должна вывести на консоль Hello World! Выберите строки, которые нужно модифицировать в программе, что бы получить правильный результат.

```
1. public class Test {
2. public Test() {
3. }
4. public static void main(String[] args) {
5. Test test = new Test();
6. String [] arr = {"H","e","l","l","o","
","w","o","r","l","d","!"};
7. String result = "";
8. int i= 0;
9. for(;;;){
10. result += arr[i++];
11. }
12. System.out.println(result);
13. }
14. }
```

Варианты ответов
1. а) Заменить строку 9 на for (i = 0; i < arr. length ;){
2. б) Заменить строку 9 на for(int int i = 0; i < arr.length;){
3. в) Заменить строку 9 на for(i = 0; i < arr.length;i++){
4. г) Заменить строку 9 на for(i = 1; i <= arr.length;i++){

Вопрос 3. Какая строка будет выдана на консоль после выполнения фрагмента кода приведенного ниже.

```
1. public class Test {
2. public Test () {
3. }
4. public static void main ( String [] args ) {
5. int i, j ;
6. lab : for ( i = 0; i < 6; i++){
7. for (j = 3; j > 1; j--){
8. if(i == j){
9. System.out.println(" " + j);
10. break lab;
```

```
11. }
12. }
13. }
14. }
15. }
```

Варианты ответов

1. а) 2345
2. б) 234
3. в) 3
4. г) 2

Вопрос 4. Какой результат следует ожидать при компиляции и запуске приведенного кода:

```
String str=new String("Java");
int i=1;
char j=3;
System.out.println(str.substring(i,j));
```

Варианты ответов

1. а) Выведено: Ja
2. б) Выведено: av
3. в) Выведено: ava
4. г) Ошибка: не существует метода substring(int,char).

Вопрос 5. Какой метод следует использовать, чтобы обнаружить позицию буквы v в строке str= "Java"?

Варианты ответов

1. а) mid (2, str)
2. б) str.charAt (2)
3. в) str.indexOf ('v')
4. г) indexOf (str,'v')

Вопрос 6. Что будет выведено в результате компиляции и запуска следующего кода:

```
String str =" ava ";
char ch ='J';
ch += str ;
System. out. println ( ch );
```

Варианты ответов

1. а) Java
2. б) ava
3. в) avaJ
4. г) J
5. д) Ошибка во время компиляции.

Вопрос 7. Что будет результатом компиляции и выполнения следующего кода?

```
StringBuffer s= new StringBuffer("You Java");
s.insert(2, "like ");
System.out.print(s);
```

Варианты ответов

1. а) Yolike u Java
2. б) You like Java
3. в) Ylike ou Java
4. г) You Java like
5. д) Ошибка компиляции: метод insert() не объявлен для класса String-Buffer

Вопрос 8. Что будет выведено при попытке компиляции и запуска этой программы:

```
public class Quest6 {
public static void main(String[] args){
int a[] = new int[]{1,2,3,};
System.out.print(a[1]);
} }
```

Варианты ответов

1. а) Ошибка компиляции: не определен размер массива
2. б) Ошибка времени выполнения
3. в) Выведено 1
4. г) Выведено 2
5. д) Ошибка компиляции: неправильная инициализация

Вопрос 9. Что будет выведено при попытке компиляции и запуска программы?

```
public class Quest8{
```

```
static int j=2;
public static void result(int i){
i *= 10;
j += 2;
}
public static void main(String[] args){
char i = ?1?;
result(i);
System.out.println(i+" "+j);
} }
```

Варианты ответов

1. а) 1 2
2. б) 10 2
3. в) Ошибка: параметр метода result() не сочетается с передаваемой переменной
4. г) 10 4
5. д) 1 4

Вопрос 10. Что будет выведено при компиляции и запуске кода?

```
public class Quest {
{System.out.print("1");}
static{System.out.print("2");}
Quest(){System.out.print("3");}
public static void main(String[] args) {
System.out.print("4");
} }
```

Варианты ответов

1. а) 34
2. б) 24
3. в) 14
4. г) 4
5. д) 1234
6. е) 234

Вопрос 11. Какие из следующих утверждений истинные?

Варианты ответов

1. а) Частные методы не могут быть перегружены
2. б) Переопределенный метод не может включать исключения не обрабатываемые в базовом классе
3. в) Методы, объявленные как final, не могут быть переопределены
4. г) Статические методы не могут быть переопределены

Вопрос 12. Что произойдет в результате компиляции и запуска кода?

```
class Base {}
class A extends Base {}
public class Quest{
public static void main(String[] args){
Base b = new Base();
A ob = (A) b;
} }
```

Варианты ответов

1. а) Ошибка во время выполнения
2. б) Ничего: компиляция и выполнение без ошибок
3. в) Ошибка во время компиляции

Вопрос 13. Что произойдет в результате компиляции и запуска кода?

```
abstract class QuestBase {
abstract void show();
static int i;
}
public class Quest2 extends QuestBase {
public static void main(String[] args){
boolean[] a = new boolean[3];
for(i = 0; i < a.length; i++)
System.out.print(? ? + a[i]);
} }
```

Варианты ответов
1. а) Ошибка времени компиляции: Quest2 должен быть объявлен как abstract
2. б) Ошибка времени выполнения: IndexOutOfBounds
3. в) будет выведено: true true true
4. г) будет выведено: false false false
5. д) Ошибка: массив a использован прежде, чем проинициализирован

Вопрос 14. Какие из объявлений корректны, если:

```
class Outer {
class Inner {
} }
```

Варианты ответов
1. а) new Outer.Inner ()
2. б) Outer.new Inner ()
3. в) new Outer.new Inner ()
4. г) new Outer().new Inner()
5. д) Outer.Inner()
6. е) Outer().Inner()
7. ж) Ни одно из приведенных.

Вопрос 15. Что будет выведено в результате компиляции и выполнения следующего кода:

```
abstract class Abstract {
abstract Abstract meth ();
}
class Owner {
Abstract meth () {
class Inner extends Abstract {
Abstract meth() {
System.out.print("Inner ");
return new Inner();
} }
return new Inner();
} }
public abstract class Quest4 {
public static void main(String a[]) {
Owner ob = new Owner();
Abstract abs = ob.meth();
abs.meth();
} }
```

Варианты ответов
1. а) Inner
2. б) Inner Inner
3. в) Inner Inner Inner
4. г) Compile time error
5. д) Runtime error
6. е) Ошибка компиляции из-за двойного объявления meth() в классе Owner

Вопрос 16. Что будет выведено в результате компиляции и выполнения этого кода?

```
import java.awt.*;
public class Quest2 extends Frame{
Quest2(){
Button yes = new Button("YES");
Button no = new Button("NO");
add(yes);
add(no);
setSize(100, 100);
setVisible(true);
}
public static void main(String[] args){
Quest2 q = new Quest();
} }
```

Варианты ответов
1. а) Две кнопки рядом, занимающие весь фрейм, YES слева и NO справа
2. б) Одна кнопка YES, занимающая целый фрейм

3. в) Одна кнопка NO, занимающая целый фрейм
4. г) Две кнопки наверху фрейма, YES и NO

Вопрос 17. Какой менеджер компоновок размещает компоненты в таблице с ячейками равного размера?
Варианты ответов
1. а) FlowLayout
2. б) GridLayout
3. в) BorderLayout
4. г) CardLayout

Вопрос 18. Какое выравнивание устанавливается по умолчанию для менеджера размещений FlowLayout?
Варианты ответов
1. а) Указывается явно
2. б) FlowLayout.RIGHT
3. в) FlowLayout.LEFT
4. г) FlowLayout.CENTER
5. д) FlowLayout. LEADING

Вопрос 19. Сколько кнопок будет выведено в апплет:

```
import java.applet.*;
import java.awt.*;
public class Quest4 extends Applet{
Button b = new Button("Yes");
public void init(){
add(b);
add(b);
add(b);
add(new Button("No"));
add(new Button("No"));
add(new Button("No"));
}}
```

Варианты ответов
1. а) 1 кнопка с надписью "Yes" и 1 кнопка с надписью "No"
2. б) 1 кнопка с надписью "Yes" и 3 кнопки с надписью "No"
3. в) 3 кнопки с надписью "Yes" и 1 кнопка с надписью "No"
4. г) 3 кнопки с надписью "Yes" и 3 кнопки с надписью "No"

Вопрос 20. Объект JCheckBox объявлен следующим образом:

```
JCheckBox ob = new JCheckBox();
```

Какая из следующих команд зарегистрирует его в блоке прослушивания событий?
Варианты ответов
1. а) addItemListener(this)
2. б) addItemListener()
3. в) ни одна из приведенных
4. г) ob. addItemListener ()
5. д) ob. addItemListener (this)

Экзамен

Вопросы к экзамену:
1. Технология Java, ее возникновение, современное состояние и развитие.
2 Структура программы Java. Библиотеки стандартных классов. Основные пакеты . Пример простой программы. Примитивные типы данных.
3. Массивы. Многомерные массивы.
4. Классы. Модификаторы доступа (инкапсуляция). Разграничение доступа в Java.
Объявление класса (заголовок + тело). Объявление полей, методов. Объявление конструкторов. Создание объекта. Инициализаторы.
5. Параметры методов. Преобразование и приведение типов. Присвоение значений.
Перегрузка методов.
6. Статические элементы. Ключевые слова this и super.Особенности их применения. Ключевое слово abstract.
7. Интерфейсы. Объявление, реализация, применение. Полиморфизм (суть, условия применения).
8. Пакет java.lang. Класс Object и его основные методы. Класс Class
9. Классы-оболочки: Integer и другие числовые классы. Их методы. Классы-оболочки Character, Boolean. Их методы. Класс Math.
10. Класс String. Конструкторы, методы. Класс StringBuffer.
11. Исключения. Иерархия исключений в Java. Генерация исключений. Создание классов исключений.

12. Перехват исключений. Вложенные блоки try. Повторная генерация исключений. Блок finally. Try с ресурсами Как использовать исключения.
13. Классы Arrays и Collections и их методы. Интерфейсы Comparator и Comparable и их применение.
14. Интерфейс List и классы его реализующие.
15. Интерфейс Set и классы его реализующие.
16. Интерфейс Map и классы его реализующие.
17. Принципы построения графического интерфейса. Парадигма MVC. Контейнеры. Иерархия компонентов и контейнеров. Компоненты. Атрибуты компонентов. Размер и позиция компонентов.
18. Структура фрейма. Создание простого фрейма. Двумерные фигуры. Иерархия фигур. Определение цвета, выбор шрифта. Вывод графических изображений.
19. Основные менеджеры компоновки.
20. Компоненты фрейма (метки, кнопки и пр.), их описание и использование.
21. Текстовые области и поля. Меню на Java.
22. Многопоточность. Состояние потока. Класс Thread и интерфейс Runnable. Создание потока. Создание нескольких потоков. Завершение потока.
23. Приоритеты потоков. Синхронизация. Взаимодействие потоков. Потоки-демоны.
24. Система ввода/вывода. Потоки данных. Байтовый ввод/вывод. Иерархия классов входных и выходных потоков. Файловый ввод и вывод. Классы фильтров ввода и вывода. Сериализация.
25. Потоки ввода символов. Консольный ввод и вывод (байтовый и символьный). Файлы произвольного доступа.

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

В КФУ действует балльно-рейтинговая система оценки знаний обучающихся. Суммарно по дисциплине (модулю) можно получить максимум 100 баллов за семестр, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов.

Для зачёта:

56 баллов и более - "зачтено".

55 баллов и менее - "не зачтено".

Для экзамена:

86 баллов и более - "отлично".

71-85 баллов - "хорошо".

56-70 баллов - "удовлетворительно".

55 баллов и менее - "неудовлетворительно".

Форма контроля	Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций	Этап	Количество баллов
Семестр 2			
Текущий контроль			
Компьютерная программа	Обучающиеся самостоятельно составляют программу на определённом языке программирования в соответствии с заданием. Программа сдаётся преподавателю в электронном виде. Оценивается реализация алгоритмов на языке программирования, достижение заданного результата.	1	20
Лабораторные работы	В аудитории, оснащённой соответствующим оборудованием, обучающиеся проводят учебные эксперименты и тренируются в применении практико-ориентированных технологий. Оцениваются знание материала и умение применять его на практике, умения и навыки по работе с оборудованием в соответствующей предметной области.	2	25
Тестирование	Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий.	3	5
Экзамен	Экзамен нацелен на комплексную проверку освоения дисциплины. Экзамен проводится в устной или письменной форме по билетам, в которых содержатся вопросы (задания) по всем темам курса. Обучающемуся даётся время на подготовку. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.		50

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

7.1 Основная литература:

Ёранссон А., Эффективное использование потоков в операционной системе Android [Электронный ресурс] / Ёранссон А. - М. : ДМК Пресс, 2015. - 304 с. - ISBN 978-5-97060-168-6 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970601686.html

Васюткина И.А., Технология разработки объектно-ориентированных программ на JAVA [Электронный ресурс]: учеб.- метод. пособие / Васюткина И.А. - Новосибирск : Изд-во НГТУ, 2012. - 152 с. - ISBN 978-5-7782-1973-1 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785778219731.html

7.2. Дополнительная литература:

Тумаков, Д.Н. Технология программирования CUDA: учебное пособие / Д.Н. Тумаков, Д.Е. Чикрин, А.А. Егорчев, С.В. Голоусов. - Казань: Изд-во Казанского университета, 2017. - 112 с. (ISBN 978-5-00019-913-8) - Режим доступа: https://shelly.kpfu.ru/e-ksu/docs/F244581044/Tekhnologiya_programmirovaniya_CUDA.pdf

Google Android: Создание приложений для смартфонов и планшетных ПК: Пособие / Голощапов А.Л. - СПб:БХВ-Петербург, 2013. - 832 с. ISBN 978-5-9775-0880-3 - Режим доступа: http://znanium.com/catalog/product/943522

Басыня Е.А., Операционные системы [Электронный ресурс]: учебно-методическое пособие / Басыня Е.А. - Новосибирск : Изд-во НГТУ, 2016. - 84 с. - ISBN 978-5-7782-3106-1 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785778231061.html

Уорбэртон Р., Лямбда-выражения в Java 8. Функциональное программирование - в массы [Электронный ресурс] / Уорбэртон Р. - М. : ДМК Пресс, 2014. - 192 с. - ISBN 978-5-94074-919-6 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785940749196.html

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Машнин Т. С. Современные Java-технологии на практике. ? СПб.: БХВ-Петербург, 2010. ? 560 с. - http://znanium.com/bookread.php?book=350724

Голощапов А. Л. Google Android: программирование для мобильных устройств. ? СПб.: БХВ-Петербург, 2010. ? 448 с. - http://znanium.com/bookread.php?book=351236

Монахов, В. В. Язык программирования Java и среда NetBeans / В. Монахов. ? 2-е изд., перераб. и доп. ? СПб.: БХВ-Петербург, 2009. ? 718 с. - http://znanium.com/bookread.php?book=351241

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лабораторные работы	Существенную роль в освоении дисциплины играют и применяемые средства разработки. В общем случае исходный код на Java можно писать и в простом текстовом редакторе, а потом выполнять компиляцию из командной строки. Но все же удобнее использовать какую-нибудь профессиональную среду разработки, например из числа этих: - IntelliJ IDEA - высокотехнологичный комплекс тесно интегрированных инструментов программирования, включая интеллектуальный редактор исходных текстов с поддержкой макросов, инструменты рефакторинга, встроенную поддержку J2EE, механизмы интеграции со средой тестирования Ant/JUnit и системами управления версиями, а также визуальный конструктор графических интерфейсов. - JBuilder - интегрированная среда разработки для языка Java, основанная на исходном коде свободной программной среде Eclipse. Необходимо отработать владение средой для выполнения заданий. После получения задания, студент должен внимательно ознакомиться с поставленной задачей, продумать пути ее решения.

Вид работ	Методические рекомендации
самостоятельная работа	Язык Java не заканчивается знаниями лишь о возможностях JSDK. Чтобы чувствовать себя комфортно необходимо знать еще десятка два сторонних библиотек, используемых Java программистами повсеместно: - Commons Lang - то, что 'забыли' включить в JDK; - Commons Math - дополнение для java.math ; - Commons Net - логическое продолжение для пакета java.net, содержит классы для работы с основными сетевыми протоколами; - Commons VFS - библиотека для абстрагирования от способа хранения файла, позволяет получить доступ до файлов по FTP, SFTP, WEBDAV, (G)ZIP и т.д.; - Commons IO - работа с вводом-выводом в Java является достаточно рутинной, библиотека Commons IO существенно расширет стандартные возможности ввода-вывода; - HttpClient - библиотека по работе с http ресурсами; - JUnit - библиотека для автоматизации процесса тестирования. Самостоятельная работа заключается в изучении этих библиотек и их использовании при решении задач.
компьютерная программа	Ниже приведён список Java-пакетов, в которых студент должен свободно ориентироваться. Знакомиться с ними рекомендуется как по JSDK-документации, так и с помощью www.exampledepot.com. Все пакеты выстроены в рекомендуемом порядке для изучения: - java.lang - основной пакет в Java, подключаемый по умолчанию; - java.io - пакет для обеспечения операцийввода-вывода; - java.util - пакет, в котором содержатся классы для работы с коллекциями: Collection, Enumeration, Set, List, Map и т.д.; - java.net - содержит основные классы для работы с сетью; - java.text - содержит все необходимое для форматирования текста; - javax.sql - содержит все необходимое по работе с базами данных; - javax.xml.*, org.w3c.dom.*, org.xml.sax.* - содержит классы для работы с XML. При написании компьютерной программы студент должен выбрать подходящий инструментарий, продумать схему проекта, построить ER - диаграмму, описать необходимые классы и интерфейсы. При этом придерживаться правила SOLID ООП.
тестирование	При подготовке к тестированию необходимо внимательно изучить выносимый на проверку материал, подключить для этого дополнительные источники, предложенные в списке литературы. Обратить особое внимание на фрагменты кодов, которые иллюстрируют изучаемый материал. Необходимо просмотреть все программы, которые были написаны во время изучения темы, а также написать новые коды, которые имеют отношение к заданной теме. Затем нужно проанализировать их на предмет возможных ошибок, уметь распознавать подобные ошибки в предложенных фрагментах кодов.
экзамен	При подготовке к экзамену необходимо внимательно изучить весь материал, подключить для этого дополнительные источники, предложенные в списке литературы. Обратить особое внимание на фрагменты кодов, которые иллюстрируют изучаемый материал. Необходимо просмотреть все программы, которые были написаны во время изучения дисциплины. Затем нужно проанализировать их на предмет возможных ошибок, уметь распознавать подобные ошибки в предложенных фрагментах кодов.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Освоение дисциплины "Технологии программирования" предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows Professional 7 Russian

Пакет офисного программного обеспечения Microsoft Office 2010 Professional Plus Russian

Браузер Google Chrome

Adobe Reader XI

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен обучающимся. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, учебно-методические комплексы, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов высшего образования (ФГОС ВО) нового поколения.

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Освоение дисциплины "Технологии программирования" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB,audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудованием имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;

- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;

- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 01.04.04 "Прикладная математика" и магистерской программе Классические и квантовые методы обработки информации .