

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и компьютерных наук

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

*Программирование на основе классов*

Кафедра дискретной математики и информатики  
факультета математики и компьютерных наук

**Образовательная программа**  
**02.03.02 - Фундаментальная информатика и информационные**  
**технологии**

Профиль подготовки  
Информатика и компьютерные науки

Уровень высшего образования  
**бакалавриат**

Форма обучения  
**очная**

Статус дисциплины: вариативная по выбору

Махачкала, 2018



### Аннотация рабочей программы дисциплины

Дисциплина “Программирование на основе классов” входит в вариативную по выбору часть образовательной программы бакалавриата по направлению 02.03.02 - Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с объектно-ориентированным программированием, созданием консольных и графических приложений.

Дисциплина нацелена на формирование следующих компетенций выпускника: ОПК-2, ПК-7.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: в форме 2-х контрольных работ в конце каждого модуля и итогового экзамена в конце семестра.

Объем дисциплины – 6 зачетных единиц, в том числе в академических часах по видам учебных занятий:

Се- местр	Учебные занятия						СРС	Форма промежуточной аттестации
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Все го	из них						
Лек- ции		Лабора- торные занятия	Практи- ческие занятия	кон- троль (экза- за- мен)	консуль- тации			
3	216	36		80	36		64	экзамен

## 1. Цели освоения дисциплины

Целями освоения дисциплины Программирование на основе классов являются овладение знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования. Формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения.

Основные задачи дисциплины: овладение методами структурного и объектно-ориентированного программирования; закрепление навыков алгоритмизации и программирования, полученных в предыдущих семестрах; создание практической базы для написания качественной выпускной квалификационной работы.

Ожидаемые результаты:

- усвоение базовых типов и операторов;
- получить знания об особенностях объектно-ориентированного программирования, изучить иерархию классов и интерфейсов;
- освоить современные методы и технологии программирования;
- получить необходимые знания о протоколах передачи информации глобальных сетей;
- приобрести навыки и умения в постановке и решении задач разработки приложений на основе классов.

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина входит в вариативную по выбору часть образовательной программы бакалавриата по направлению 02.03.02 - Фундаментальная информатика и информационные технологии и изучается в соответствии с графиком учебного процесса в 3 семестре. Изучение предмета производится в течение одного семестра и заканчивается экзаменом.

Дисциплина опирается на знания, полученные в 1 семестре в процессе изучения дисциплин «Основы программирования», во 2 семестре «Языки программирования». В свою очередь, на материал данной дисциплины опираются дисциплины «Java-программирование», «Программная инженерия» и «Введение в Case-технологии»; знания, умения и навыки, полученные при изучении данной дисциплины, используются, закрепляются и развиваются при проведении учебной практики, выполнении курсовой и выпускной квалификационной работы.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения)

Компетенции	Наименование компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ОПК-2	способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современ-	<b>Знать:</b> основы современных методов программирования, механизмы взаимодействия web-сервера и клиента. <b>Уметь:</b> применять методы объектно-ориентированного, визуального и событийно-управляемого программирования. <b>Владеть:</b> начальными навыками разработки программ с применением методов объектно-ориентированного, визуального и событийно-управляемого программирования навы-

	ные профессиональные стандарты информационных технологий	ками работы в различных программных средах.
ПК-7	способность разрабатывать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов систем информационных технологий, а также методы и механизмы оценки и анализа функционирования средств и систем информационных технологий	<p><b>Знать:</b> элементы языка высокого уровня: типы, операторы, иерархию классов и интерфейсов; основы алгоритмизации, основы оптимального представления входных данных с использованием наиболее подходящих структур.</p> <p><b>Уметь:</b> разрабатывать простые алгоритмы и воплощать в программы, использовать современные операционные системы и оболочки при создании программных приложений, разрабатывать проектную и программную документацию.</p> <p><b>Владеть:</b> развитыми навыками объявления и применения ссылочных типов (строки, массивы, списки, коллекции и др.)</p>

#### 4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 6 зачетные единицы, 216 академических часов: 36ч. лекций, 80 ч. практ., 64 – СРС, 36 – контроль в виде экзамена.

4.2. Структура дисциплины

Структура и содержание дисциплины «Программирование на основе классов»

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра)  Форма промежуточной аттестации (по семестрам)
				Всего	Лекции	Практические	Лабораторные занятия	Самостоятельная раб.	Контроль самост. раб.	
	Модуль 1. Основы объектно-ориентированных языков.									
1	Тема 1.1. Введение в программирование на. Инструментальные средства разработки.	3	1	8	2	4		4	2	Устный опрос
2	Тема 1.2. Базовые элементы языка. Типы.	3	2	12	2	6		3	1	Устный опрос

	Тема 1.3. Управляющие структуры	3	3	16	2	6		3	1	Устный опрос
	Итого по модулю 1			36	6	16		10	4	Контрольная работа №1
<b>Модуль 2. Принципы ОПОП</b>										
4	Тема 2.1. Принципы ОПОП.	3	4	12	2	4		4	2	Устный опрос
	Тема 2.2. Классы и объекты.		5	12	2	6		3	1	Устный опрос
5	Тема 2.3. Работа со структурированными типами	3	6	12	2	6		3	1	Устный опрос
	Итого по модулю 2			36	6	16		10	4	Контрольная работа №2
<b>Модуль 3. Программирование классов</b>										
1	Тема 3.1. Создание CGI-программ.	3	7	12	2	6		3	1	Устный опрос
2	Тема 3.2. Компиляция и выполнение CGI-программ.	3	8	12	2	4		3	1	Устный опрос
3	Тема 3.3. Средства пользовательского интерфейса. Компоненты и контейнеры.	3	9	12	2	6		3	1	Устный опрос
	Итого по модулю 3			36	6	16		9	3	Контрольная работа №3
<b>Модуль 4. Обработка событий</b>										
4	Тема 4.1. События. Обработка событий от компонент.	3	10	12	2	6		3	1	Устный опрос
5	Тема 4.2. Графические примитивы. Работа с графикой.	3	11	12	2	6		3	1	Устный опрос
6	Тема 4.3 Создание, выполнение и синхронизация потоков. Многопоточность. Приоритеты пото-	3	12-13	12	4	4		3	1	Устный опрос

	ков.									
	Итого по модулю 4			<b>36</b>	<b>8</b>	16		9	3	Контрольная работа №4
	Модуль 5. Создание приложений									
7	Тема 5.1. Создание потоков, связанных с файлами. Файловый ввод/вывод.	3	14	12	2	4		4	2	Устный опрос
8	Тема 5.2 Программирование меню приложений.	3	15-16	12	4	6		2		Устный опрос
9	Тема 5.3. Особенности создания сетевых приложений.	3	17-18	12	4	6		2		Устный опрос
	<i>Итого по модулю 5:</i>			36	10	16		8	2	Контрольная работа №5
	<i>Модуль 6. Подготовка к экзамену</i>			36						Экзамен (36 ч.)
	<b>ИТОГО:</b>			216	36	80		47	17	

#### 4.3. Содержание дисциплины, структурированное по разделам и темам

4.3.1. Содержание лекционных занятий по дисциплине, структурированное по разделам и темам

##### **Модуль 1. Основы объектно-ориентированных языков.**

##### **Тема 1.1. Введение в ОПОП. Инструментальные средства разработки.**

Документация, литература.

Понятие байт-кода.

Пакетные средства, интегрированные средства разработки.

##### **Тема 1.2. Базовые элементы языка. Типы.**

Структура программы.

Типы данных. Идентификаторы, константы, переменные.

Преобразование типов.

Массивы.

##### **Тема 1.3. Базовые элементы языка. Операторы.**

Операторы выбора, цикла, перехода.

Операторы и блоки. Оператор if-else. Оператор switch. Цикл while и do-while. Оператор for. Метки. Оператор break. Оператор continue. Оператор return.

Использование функций.

Модуль 2. Принципы ОПОП

##### **Тема 2.1. Основы объектно-ориентированного программирования.**

Тема 2.1. Принципы ОПОП.

Основные понятия.

Инкапсуляция.

Наследование.

Полиморфизм.

Тема 2.2. Классы и объекты.

Классы, интерфейсы, абстрактные классы.

Реализация классов.

Создание объекта, оператор new.

Правила доступа к данным и методам объекта.

Переопределение методов.

Метод-конструктор.

Тема 2.3. Работа со структурированными типами

Классы String, StringBuffer.

Равенство строк. Сравнение. Упорядочение.

Методы работы со строками.

Методы indexOf, lastIndexOf.

Типы исключений.

Вложенные операторы try.

Блок finally. Операторы throw, catch. Подклассы Exception.

### **Модуль 3. Программирование классов**

Тема 3.1. Создание CGI-программ.

Создание апплета на Java.

Параметры апплета.

Исходный текст апплета.

Методы init, destroy, start, stop.

Тема 3.2. Компиляция и выполнение CGI-программ.

Компиляция и выполнение CGI-программ.

Параметры CGI-программ.

Тема 3.3. Средства пользовательского интерфейса. Компоненты и контейнеры.

Кнопки, переключатели.

Списки класса Choice.

Списки класса List.

Текстовое поле классов Label, TextField.

Многострочное текстовое поле класса TextArea.

### **Модуль 4. Обработка событий**

Тема 4.1. События. Обработка событий от компонент.

Обработка событий.

Метод handleEvent.

Класс MouseEvent и интерфейс MouseListener.

Движение мыши, перетаскивание.

Событие от клавиатуры.

Обработка событий, инициированных компонентами.

Тема 4.2. Графические примитивы. Работа с графикой.

Загрузка и рисование растрового изображения.

Класс Image. Метод paint.

Координаты, цвет, шрифт.

Графические элементы. Класс Graphics. Метод setColor.

Просмотр изображения в апплете.

Видео в окне апплета.

Загрузка и проигрывание звуковых файлов.

Тема 4.3 Создание, выполнение и синхронизация потоков. Многопоточность. Приоритеты потоков.

Процессы, потоки и приоритеты. Создание потоков.

Синхронизация. Методы synchronized. Операторы synchronized.

Методы wait и notify. Планирование потоков.  
Взаимная блокировка. Приостановка потоков. Прерывание потока.  
Завершение работы потока. Завершение приложения.  
Использование Runnable. Ключевое слово volatile.  
Безопасность потоков и Thread Group. Отладка потоков.  
Реализация многопоточности в Java.  
Модуль 5. Создание приложений  
Тема 5.1. Создание потоков, связанных с файлами. Файловый ввод/вывод  
Потоки ввода и вывода.  
Файловый ввод/вывод.  
Построчный и побайтный ввод/вывод.  
Тема 5.2 Программирование меню приложений.  
Классы Menubar, Menu и MenuItem.  
Создание меню в окне типа Frame.  
Отображение меню.  
Тема 5.3. Особенности создания сетевых приложений.  
Адрес IP и класс InetAddress.  
Универсальный адрес ресурсов URL.  
Класс URL в библиотеке классов Java.  
Связь приложений Java с расширениями сервера Web.  
Приложения SocketServ и SocketClient.  
4.3.2. Содержание практических занятий по дисциплине, структурированное по разделам и темам

## **Модуль 1. Основы объектно-ориентированных языков.**

### **Тема 1.1. Введение в ОПОП. Инструментальные средства разработки.**

Понятие байт-кода. Java-машина. Кроссплатформенные программы.

Пакетные средства, интегрированные средства разработки.

### **Тема 1.2. Базовые элементы языка. Типы.**

Структура программы, простые типы, формат их представления в памяти.

Преобразование типов. Особенности массивных типов.

### **Тема 1.3. Базовые элементы языка. Операторы.**

Условный оператор и его развитие – оператор выбора. Различия и взаимозаменяемость операторов цикла. Минусы оператора перехода. Операторы прерывания, продолжения, возврата. Использование функций.

Модуль 2. Принципы ОПОП

### **Тема 2.1. Основы объектно-ориентированного программирования.**

Тема 2.1. Принципы ОПОП.

Три основных принципа ОПОП. Как инкапсуляция разграничивает права доступа. Применение идеи наследования. Примеры полиморфизма.

Тема 2.2. Классы и объекты.

Классы, интерфейсы, абстрактные классы. Реализация классов и создание объекта. Правила доступа к данным и методам объекта. Переопределение методов. Конструктор.

Тема 2.3. Работа со структурированными типами

Классы String, StringBuffer. Равенство строк. Сравнение. Упорядочение.

Методы работы со строками. Методы indexOf, lastIndexOf.

Типы исключений. Вложенные операторы try. Блок finally. Операторы throw, catch. Подклассы прерываний.

## **Модуль 3. Программирование классов**

Тема 3.1. Создание CGI-программ.

Создание апплета на Java, параметры, исходный текст. Методы init, destroy, start, stop.

Тема 3.2. Компиляция и выполнение CGI-программ.

Компиляция и выполнение CGI-программ. Параметры CGI-программ.

Тема 3.3. Средства пользовательского интерфейса. Компоненты и контейнеры.

Кнопки, переключатели. Списки классов Choice и List. Текстовое поле классов Label, TextField. Многострочное текстовое поле класса TextArea.

#### **Модуль 4. Обработка событий**

Тема 4.1. События. Обработка событий от компонент.

Обработка событий. Метод `handleEvent`. Класс `MouseEvent` и интерфейс `MouseListener`.

События от мыши и клавиатуры. Обработка событий, инициированных компонентами.

Тема 4.2. Графические примитивы. Работа с графикой.

Загрузка и отображения растрового изображения. Класс `Image`. Метод `paint`.

Координаты, цвет, шрифт. Графические элементы. Класс `Graphics`. Метод `setColor`. Просмотр изображения в апплете. Видео в окне апплета.

Загрузка и проигрывание звуковых файлов.

Тема 4.3 Создание, выполнение и синхронизация потоков. Многопоточность. Приоритеты потоков.

Процессы, потоки и приоритеты. Создание потоков.

Синхронизация. Методы `synchronized`. Операторы `synchronized`.

Методы `wait` и `notify`. Планирование потоков. Взаимная блокировка. Приостановка потоков. Прерывание потока. Завершение работы потока. Завершение приложения. Использование `Runnable`. Ключевое слово `volatile`. Безопасность потоков и `Thread Group`. Отладка потоков. Реализация многопоточности в Java.

Модуль 5. Создание приложений

Тема 5.1. Создание потоков, связанных с файлами. Файловый ввод/вывод

Потоки ввода и вывода. Файловый ввод/вывод. Построчный и побайтный ввод/вывод.

Тема 5.2 Программирование меню приложений.

Классы `MenuBar`, `Menu` и `MenuItem`. Создание меню в окне типа `Frame`. Отображение меню.

Тема 5.3. Особенности создания сетевых приложений.

Адрес IP и класс `InetAddress`. Универсальный адрес ресурсов URL. Класс URL в библиотеке классов Java. Связь приложений Java с расширениями сервера Web. Приложения `SocketServ` и `SocketClient`.

### **5. Образовательные технологии**

Процесс изложения учебного материала сопровождается систематическими (на каждом занятии) компьютерными презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного оборудования, предусмотрено широкое использование в учебном процессе активных и интерактивных форм проведения занятий.

Студентам регулярно сообщаются те из формулировок задач, предложенных на собеседовании в ведущие IT-компании (Microsoft, Apple, Google, Uber, Amazon.com и др.), которые соответствуют текущим темам.

. Предусмотрено регулярное общение и консультации с представителями российских и зарубежных компаний (из числа выпускников кафедры) по электронной почте и по скайпу.

### **6. Учебно-методическое обеспечение самостоятельной работы студентов**

Для обеспечения самостоятельной работы используется разработанный на кафедре пакет заданий и методических указаний. Самостоятельная работа студентов складывается из проработки лекционного материала и материалов форумов программирования, материала учебника, решения всех заданий из индивидуальных заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных форма контроля.

Пример задания на самостоятельную работу.

**Задание**

Выведите на экран все двузначные члены последовательности  $2a_{n-1}+50$ , где  $a_1 = -26$ .  
Создайте массив из 11 случайных целых чисел из отрезка  $[-1;1]$ , выведите массив на экран в строку. Определите какой элемент встречается в массиве чаще всего и выведите об этом сообщение на экран. Если два каких-то элемента встречаются одинаковое количество раз, то не выводите ничего.

Создайте класс углов, отложенных против часовой стрелки от положительного направления оси абсцисс, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий задают ли углы перпендикулярные прямые.

С использованием построенного класса создайте угол в  $10^\circ$  и второй угол в  $280^\circ$ . Проверьте с помощью созданного метода задают ли углы перпендикулярные прямые и если да, то выведите соответствующее сообщение на экран.

## 7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

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

Код компетенции из ФГОС ВО	Наименование компетенции из ФГОС ВО	Планируемые результаты	Процедура освоения
ОПК-2	способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий	Знать: основы языка Java, алгоритмические и программные решения в области системного и прикладного программирования.	Изучение тем модулей 1-3. Выполнение тестовых заданий на каждом лабораторном занятии для проверки усвоения самостоятельных заданий.
		Уметь: использовать типы, операции, управляющие структуры и визуальные компоненты Java в разработке приложений, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей.	Составление программ с использованием различных типов и с применением всех структур управления
		Владеть: навыками объектно-ориентированного программирования, создания консольных программ, и GUI-приложений и апплетов.	Отладка и тестирование программ по темам модулей 1-3.
ПК-7 ПК-7	способность разрабатывать и реализо-	Знать: элементы языка Java: типы, операторы, иерархию классов и ин-	Проработка тем модулей 4-5

вывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов систем информационных технологий, а также методы и механизмы оценки и анализа функционирования средств и систем информационных технологий	терфейсов; основы алгоритмизации, основы оптимального представления входных данных с использованием наиболее подходящих структур.	
	Уметь: разрабатывать простые алгоритмы и воплощать в программы, использовать современные операционные системы и оболочки при создании программных приложений, разрабатывать проектную и программную документацию.	Составление программ с использованием в отдельности файлов, массивов, строк, списков и коллекций
	Владеть: развитыми навыками объявления и применения ссылочных типов (строки, массивы, списки, коллекции и др.)	Составление, отладка и тестирование программ с использованием нескольких структурированных типов (с обязательным присутствием файловых данных),

## 7.2. Типовые контрольные задания

### Вариант 1

1. В переменной  $n$  хранится четырехзначное число. Создайте программу, вычисляющую и выводящую на экран сумму цифр  $n$ .
2. Создайте массив из 15 случайных целых чисел из отрезка  $[0;9]$ . Выведите массив на экран. Подсчитайте сколько в массиве чётных элементов и выведете это количество на экран на отдельной строке.
3. Создайте класс окружностей на плоскости, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий имеют ли две окружности равную площадь.

С использованием построенного класса создайте две окружности: одну с центром в  $(0;0)$  и радиусом 12, а вторую с центром в  $(3;5)$  и радиусом 11. Проверьте с помощью созданного метода равна ли их площадь и если равна, то выведите соответствующее сообщение на экран.

### Вариант 2

1. В переменных  $q$  и  $w$  хранятся два натуральных числа. Создайте программу, выводящую на экран результат деления  $q$  на  $w$  с остатком. Пример вывода программы (для случая, когда в  $q$  хранится 21, а в  $w$  хранится 8):  $21 / 8 = 2$  и 5 в остатке.
2. Создайте массив из 8 случайных целых чисел из отрезка  $[1;10]$ . Выведите массив на экран в строку. Замените каждый элемент с нечётным индексом на ноль. Снова выведете массив на экран на отдельной строке.

3. Создайте класс отрезков на координатной плоскости, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий равна ли длина двух отрезков.

С использованием построенного класса создайте два отрезка: один от точки (1;1) до точки (2;2) и второй отрезок от точки (-3;0) до точки (1;1). Проверьте с помощью созданного метода равна ли их длина и если равна, то выведите соответствующее сообщение на экран.

### Вариант 3

1. Создать программу, которая будет проверять попало ли случайно выбранное из отрезка [5;155] целое число в интервал (25;100) и сообщать результат на экран.
2. Создайте массив из 8 случайных целых чисел из отрезка [1;10]. Выведите массив на экран в строку. Замените каждый элемент с нечётным индексом на ноль. Снова выведете массив на экран на отдельной строке.
3. Создайте класс прямоугольных треугольников, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий являются ли два треугольника подобными.

С использованием построенного класса создайте два треугольника: один с катетами 3 и 4 и второй с катетами 5 и 6. Проверьте с помощью созданного метода подобны ли треугольники и если да, то выведите соответствующее сообщение на экран.

### Вариант 4

1. Создать программу, выводящую на экран случайно сгенерированное трёхзначное натуральное число и его наибольшую цифру.
2. Создайте массив из 4 случайных целых чисел из отрезка [10;99], выведите его на экран в строку. Определить и вывести на экран сообщение о том, является ли массив строго возрастающей последовательностью.
3. Создайте класс комплексных чисел, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, вычисляющий модуль суммы двух комплексных чисел.

С использованием построенного класса создайте два комплексных числа:  $3i+1$  и  $2i-1$ . Вычислите с помощью метода и выведите на экран модуль их суммы.

### Вопросы

Вопросы экзамена по курсу «Программирование на основе классов» для студентов 2 курса ФМиКН направления ФИИТ, 3 семестр.

1. Принципы объектно-ориентированного программирования.
2. Понятие кроссплатформенности.
3. Идентификаторы. Комментарий. Управляющие символы.
4. Типы данных. Примитивные типы.
5. Типы данных. Ссылочные типы. Массивы.
6. Типы данных. Ссылочные типы. Классы. Интерфейсы.
7. Логические операции. Преобразование типа. Переменные.
8. Операторы. Логические, присваивания. Приоритеты.

9. Операторы. Условные, выбора.
10. Операторы цикла. Три инструкции перехода.
11. Свойства ОПОП.
12. Классы. Передача параметров по значению и по ссылке. Ссылка this.
13. Модификаторы доступа. Типы методов: перегруженные, переопределенные. Ключевое слово super.
14. Пакеты. Включение пакета, класса пакета. Пакет AbstractWindowToolkit, GUI.
15. Контейнеры: фрейм, панель.
16. Компоненты GUI: Label, TextField, TextArea.
17. Компоненты GUI: Button, Checkbox, радиокнопки, списки.
18. Типы макетов.
19. Апплеты - создание, жизненный цикл, вывод апплета.
20. Апплеты – вывод изображений, передача параметров.
21. Классы Graphics, Colors, Fonts.
22. Обработка исключений. Классы ErrorException.
23. Ключевые слова: try, catch, throw, throws, finally.
24. Потоки, способы создания потоков.
25. Состояния потоков, два типа потоков.
26. Пакеты, управление доступом.

### **7.3. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Общий результат складывается из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий – 10 баллов,
- выполнение текущих практических заданий – 40 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 50 баллов,
- письменная контрольная работа - 50 баллов.

### **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

#### **Основная литература:**

1. Джошуа Блох Java. Эффективное программирование [Электронный ресурс]/ Джошуа Блох— Электрон. текстовые данные. — Саратов: Профобразование, 2017.— 310 с.— Режим доступа: <http://www.iprbookshop.ru/64057.html>.— ЭБС «IPRbooks».
2. Вязовик Н.А. Программирование на Java [Электронный ресурс]/ Вязовик Н.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 603 с.— Режим доступа: <http://www.iprbookshop.ru/73710.html>.— ЭБС «IPRbooks»3.
3. Бен-Ари М. Языки программирования. Практический сравнительный анализ. Изд. “Мир”. М., 2000.
4. Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5 (6-е издание). Издательство: Вильямс, 2013.
5. Джозеф Албахари, Бен Албахари. C# 5.0. Справочник. Полное описание языка (5-е издание). Издательство: Вильямс, 2013. - 1054 стр.

7. Патрик Ноутон, Герберт Шилдт. Java 2. Наиболее полное руководство. Санкт-Петербург. “БХВ-Петербург” 2013, 1050 стр.

#### **Дополнительная литература:**

1. Мухаметзянов Р.Р. Основы программирования на Java [Электронный ресурс]: учебное пособие/ Мухаметзянов Р.Р.— Электрон. текстовые данные.— Набережные Челны: Набережночелнинский государственный педагогический университет, 2017.— 114 с.— Режим доступа: <http://www.iprbookshop.ru/66812.html>.— ЭБС «IPRbooks»

2. В. Пестриков, А. Маслобоев. Delphi на примерах. СПб.: БХВ-Петербург, 2005.

#### **9. Перечень рекомендуемых ресурсов информационно-телекоммуникационной сети «Интернет»**

<https://ru.wikipedia.org/wiki/Java>, <http://citforum.ru/>, <http://www.compdoc.ru/>,  
<http://www.emanual.ru/>

#### **10. Методические указания для обучающихся по освоению дисциплины**

1) Помимо выполнения заданий на практических занятиях рекомендуется самостоятельно решить упражнения, предложенных к каждой лекции.

2) Самостоятельная работа студентов заключается в решении всех разобранных на занятиях упражнений, материала учебника и соответствующих форумов интернет, решения всех заданий из индивидуальных практических заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных отчетов и экзамена и дополнительной работы в компьютерном классе самостоятельно.

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

Для проведения полноценных занятий необходимо следующее программное обеспечение: Операционная система Windows 7, 8.1 и 10, JDK, Microsoft Visual Studio Express, NetBeans, Ubuntu Linux.

#### **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Лекции по дисциплине читаются в классе, оборудованном проектором, к каждому занятию имеются презентации, лабораторные работы проходят в компьютерном классе, оборудованном необходимым аппаратными и программными средствами. Часть лекций предоставляется студенту в электронном формате. Практические занятия проводятся в компьютерных классах с современным аппаратным и программным обеспечением. На сайте кафедры размещаются учебные пособия и презентации к лекции.