

Скоков
Роман
Юрьевич
Ректор "
ВИЭПП"

2026.02.27
15:46:55
+03'00'

Приложение 2 к рабочей программе дисциплины
«Визуальное программирование на языке Java»

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛЖСКИЙ ИНСТИТУТ ЭКОНОМИКИ, ПЕДАГОГИКИ И ПРАВА»

Факультет информационной безопасности и права

Фонд оценочных средств
по дисциплине
«Визуальное программирование на языке»

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль) образовательной программы:
Общий правовой профиль

Уровень высшего образования:
бакалавриат

Квалификация выпускника:
«бакалавр»

Волжский 2026 г.

Содержание

1	Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы, описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	3
2	Методические материалы и типовые контрольные задания, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	6
2.1	Типовые контрольные задания, используемые для проведения входного контроля.....	6
2.2	Типовые контрольные задания, используемые для промежуточной аттестации по дисциплине.....	7
2.3	Методические материалы и типовые контрольные задания, используемые для текущего контроля по дисциплине.....	7
2.3.1	Методические материалы, используемые для текущего контроля знаний по дисциплине.....	7
2.3.2	Вопросы, выносимые на самостоятельное изучение.....	8
2.3.3	Задания для самостоятельной работы.....	8
2.3.4	Тесты по дисциплине.....	8
2.3.5	Типовые задания.....	9
2.3.6	Задания для контрольной работы.....	10
2.3.7	Методика проведения лекции-дискуссии.....	10
2.3.8	Методика проведения интерактивного решения задач.....	12
3	Методические материалы, определяющие процедуры оценивания знаний, умений, навыков по дисциплине.....	15
3.1	Балльно-рейтинговая система оценки успеваемости по дисциплине.....	15

1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы, описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Перечень компетенций	Показатели оценивания компетенций	Критерии оценивания компетенций	Этапы формирования компетенций	Шкала оценивания
ПК-1. Способен осуществлять руководство разработкой программного кода	методы и приемы формализации и алгоритмизации задач, нормативные документы определяющие требования к оформлению программного кода, нотации и программные продукты для графического отображения алгоритмов; методологии и жизненные циклы разработки ПО; принципы архитектуры веб-приложений; стандарты именования, документирования и рефакторинга кода;	- знает принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция); методологии и парадигмы проектирования; стандарты оформления кода; жизненный цикл ПО; - умеет проектировать архитектуру ПО на основе ООП; выбирать оптимальные паттерны проектирования; оценивать качество программного кода; организовывать работу команды разработчиков; - владеет инструментами проектирования (UML, диаграммы); методами рефакторинга; техниками code review; средами разработки (IDE)	Начальный	удовлетворительно (60–74 балла)
		- знает принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция); методологии и парадигмы	Основной	хорошо (75–89 баллов)

		проектирования; стандарты оформления кода; жизненный цикл ПО; - умеет проектировать архитектуру ПО на основе ООП; выбирать оптимальные паттерны проектирования; оценивать качество программного кода; организовывать работу команды разработчиков; - владеет инструментами проектирования (UML, диаграммы); методами рефакторинга; техниками code review; средами разработки (IDE)		
		- знает принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция); методологии и парадигмы проектирования; стандарты оформления кода; жизненный цикл ПО; - умеет проектировать архитектуру ПО на основе ООП; выбирать оптимальные паттерны проектирования; оценивать качество программного кода; организовывать работу команды разработчиков; - владеет инструментами проектирования (UML, диаграммы); методами	Завершающий	отлично (90–100 баллов)

		рефакторинга; техниками code review; средами разработки (IDE)		
--	--	---	--	--

2.3 Методические материалы и типовые контрольные задания, используемые для текущего контроля по дисциплине

2.3.1 Методические материалы, используемые для текущего контроля знаний по дисциплине

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Контрольный опрос	Контрольный опрос – это метод оценки уровня освоения компетенций, основанный на непосредственном (беседа, интервью) или опосредованном (анкета) взаимодействии преподавателя и студента. Источником контроля знаний в данном случае служит словесное или письменное суждение студента	Примерный перечень вопросов к экзамену Вопросы, выносимые на самостоятельное изучение Задания для самостоятельной работы
Собеседование	Собеседование – это один из методов контрольного опроса, представляющий собой относительно свободный диалог между преподавателем и студентом на заданную тему	Примерный перечень вопросов к экзамену Вопросы, выносимые на самостоятельное изучение Задания для самостоятельной работы
Тестовые задания	Тестирование - удовлетворяющая критериям исследования эмпирико-аналитическая процедура оценки уровня освоения компетенций студентами	Тесты по дисциплине
Самостоятельное выполнение типовых заданий	Метод, при котором обучающиеся приобретают навыки творческого мышления, самостоятельного решения проблем теории и практики.	Типовые задания
Контрольная работа	Эффективный метод проверки знаний обучающихся, полученных ими на определённом этапе. Основная задача контрольных работ - выявить, какие изученные темы вызывают затруднения и в результате искоренить недостатки	Примерный перечень вопросов к экзамену Вопросы, выносимые на самостоятельное изучение
Круглый стол	Метод, позволяющий включить обучающихся в процесс обсуждения спорного вопроса, проблемы и оценить их умение аргументировать собственную точку зрения.	Перечень дискуссионных тем для проведения круглого стола
Выполнение домашнего задания	Продукт самостоятельной интеллектуальной работы студента основной целью, которого является расширение объёма приобретаемых знаний по данной теме.	Вопросы, выносимые на самостоятельное изучение

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Написание реферата	Продукт самостоятельной работы обучающегося, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы	Методика написания реферата Темы рефератов

2.1 Типовые контрольные задания, используемые для проведения входного контроля

1. Классификация программного обеспечения.
2. Назовите известные вам интерпретаторы и компиляторы.
3. Перечислите базовые типы данных языка C/C++.
4. Нарисуйте схему алгоритма вычисления суммы бесконечного ряда для нахождения e .
5. Приведите таблицы истинности для основных логических операций.
6. Назовите основные алгоритмические конструкции структурного программирования.
7. Запишите в дополнительном коде целое число -5.
8. Нарисуйте схему алгоритма вычисления максимального и минимального элементов одномерного массива.

2.2. Типовые контрольные задания, используемые для промежуточной аттестации по дисциплине

1. Разработать программу для выполнения арифметической операции над двумя числами. Код должен позволять диалоговыми средствами последовательно ввести первое число, затем - второе число, затем - арифметическую операцию. Программа должна анализировать введенные данные на корректность, запрашивать у пользователя подтверждение на продолжение работы, если данные не корректны и завершать работу в случае отсутствия подтверждения. В результирующем диалоговом окне вывести результат вычисления введенной арифметической операции над введенными числами.

2. Дано целое число $n = 100$, а также первый член $a = 154$ и разность $d = 3$ арифметической прогрессии. Сформировать и вывести массив размера n , содержащий n первых членов данной прогрессии: $a, a+d, a+2*d, a+3*d, \dots$.

2.2.1. Примерный перечень вопросов к зачету

Теоретические вопросы

1. Парадигмы программирования.
2. Составные части объектного подхода. Абстрагирование.
3. Составные части объектного подхода. Инкапсуляция.
4. Составные части объектного подхода. Модульность.
5. Составные части объектного подхода. Иерархия.
6. Составные части объектного подхода. Типизация.
7. Составные части объектного подхода. Параллелизм.

Практические вопросы

Объектно-ориентированная модель. Понятие объекта. Типы данных Java.

1. Строгая типизация Java.
2. Базовые типы языка. Строки и литералы.
3. Преобразования типов в выражениях.

4. Особенности инициализации массивов. Операторы управления памятью.
5. Структура объявления класса.
6. Доступ к членам класса.
7. Спецификаторы доступа для обеспечения инкапсуляции.
8. Работа с массивами объектов.
9. Логические тип и перечисления.
10. Представление объектов и классов.
11. Поля данных класса как механизм реализации состояния объекта.
12. Функции члены класса как механизм реализации поведения объекта.
13. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности.
14. Наследование как средство организации иерархий классов.
15. Одиночное наследование.
16. Множественное наследование.

2.2.2 Примерное задание на зачет

Задание: Создать статические методы, вычисляющие факториал натурального числа, как рекурсивным, так и итерационным способами. Сравнить быстродействие этих методов, подсчитав, сколько умножений выполняется в первом и во втором случаях при вычислении факториалов 6, 7 и 8.

Создать класс векторов n -мерного пространства в декартовой системе координат. Реализовать операции сложения и вычитания векторов, умножения вектора на скаляр, вычисления длины вектора, проверки коллинеарности двух векторов.

2.3 Методические материалы и типовые контрольные задания, используемые для текущего контроля по дисциплине

2.3.1 Методические материалы, используемые для текущего контроля знаний по дисциплине

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде	Наименование оценочного средства
Контрольный опрос	Метод оценки уровня освоения компетенций на основе непосредственного взаимодействия преподавателя и студента	Примерный перечень вопросов к зачету, вопросы для самостоятельного изучения	Контрольный опрос
Собеседование	Относительно свободный диалог между преподавателем и студентом на заданную тему	Вопросы к практическим занятиям, задания для самостоятельной работы	Собеседование
Тестовые задания	Эмпирико-аналитическая процедура оценки	Тесты по дисциплине (раздел 2.3.4)	Тестовые задания

	уровня освоения компетенций		
Самостоятельное решение задач	Метод приобретения навыков творческого мышления и самостоятельного решения проблем	Типовые задания (раздел 2.3.5), задания для контрольной работы	Самостоятельное решение задач
Лекция-дискуссия	Оценочное средство для включения обучающихся в обсуждение спорных вопросов	Методика проведения лекции-дискуссии (раздел 2.3.7)	Лекция-дискуссия
Интерактивное решение задач	Метод модерации с участием всех обучающихся под руководством преподавателя	Методика проведения интерактивного решения задач (раздел 2.3.8)	Интерактивное решение задач

2.3.2 Вопросы, выносимые на самостоятельное изучение

Теоретический блок:

1. Какие есть типы данных в Java?
3. Чем отличается объект от примитивных типов данных?
4. В чем разница передачи параметров по ссылке и по значению?
5. Что такое JVM, JDK, JRE?
6. Зачем используют JVM?
7. Что такое bytecode?
8. Какие признаки JavaBean?
9. Что такое OutOfMemoryError?
10. Что такое стектрейс? Как его получить?
11. Назовите все методы класса object.
12. В чем отличие между try-with-resources и try-catch-finally при работе с ресурсами?
13. Что такое конструкторы? Какие типы знаете?
14. Что такое побитовые операции?
15. Объекты каких стандартных классов immutable в Java?
16. Дайте краткую характеристику immutable object. Зачем они нужны?
17. Как создать immutable object?
18. Какие преимущества immutable object перед обычными объектами?

Практический блок:

1. Дайте краткую характеристику class object в Java.
2. Для чего используют Equals and hashCode в Java?
3. Расскажите о контракте между Equals and hashCode в Java?
4. Какие условия выдвигаются касательно переопределения соглашения при переопределении Equals?
5. Что будет, если не переопределить Equals and hashCode?
6. Какие значения мы получим, если у нас не перераспределены Equals and hashCode?
7. Почему симметричность выполняется, только если x.equals(y) возвращает true?
8. Что такое коллизия в hashCode? Как с ней бороться?
9. Что будет, если элемент, участвующий в контракте с hashCode, меняет свое значение?

10. Напишите методы Equals and hashCode для класса Student, который состоит из полей String name и int age.

11. В чем разница применения if (obj instanceof Student) и if (getClass() == obj.getClass())?

12. Дайте краткую характеристику метода clone().

13. В чем заключается особенность работы метода clone() с полями объекта типа-ссылки?

2.3.3 Задания для самостоятельной работы

1. Напишите программу, которая выводит на экран числа от 1 до 10..

2. Найдите факториал заданного числа

3. Отсортируйте массив по значению в порядке возрастания и убывания.

4. Напишите на Java простой код, который переводит целое число в строку, при том что его можно применить в любой системе счисления.

5. Напишите программу, которая инвертирует массив (меняет порядок элементов на противоположный).

2.3.4 Тесты по дисциплине

1. Given: public class Sequence { Sequence() { System.out.print("x "); } public static void main(String[] args) { new Sequence().go(); } void go() { System.out.print("y "); } } What is the result? A) x y B) y x C) x c D) y c

2. Given: public class MyStuff { MyStuff(String n) { name = n; } String name; public static void main(String[] args) { MyStuff m1 = new MyStuff("guitar"); MyStuff m2 = new MyStuff("tv"); System.out.println(m2.equals(m1)); } public boolean equals(Object o) { MyStuff m = (MyStuff) o; if(m.name != null) return true; return false; } } What is the result? A) The output is "true" and MyStuff fulfills the Object.equals() contract. B) The output is "false" and MyStuff fulfills the Object.equals() contract. C) The output is "true" and MyStuff does NOT fulfill the Object.equals() contract. D) The output is "false" and MyStuff does NOT fulfill the Object.equals() contract E) Compilation fails

3. Given: public class MyLoop { public static void main(String[] args) { String[] sa = {"tom ", "jerry "}; for(int x = 0; x < 3; x++) { for(String s: sa) { System.out.print(x + " " + s); if(x == 1) break; } } } } What is the result? A) 0 tom 0 jerry 1 tom B) 0 tom 0 jerry 1 tom 1 jerry C) 0 tom 0 jerry 2 tom 2 jerry D) 0 tom 0 jerry 1 tom 2 tom 2 jerry E) 0 tom 0 jerry 1 tom 1 jerry 2 tom 2 jerry

4. Given: class Feline { public String type = "f "; public Feline() { System.out.print("feline "); } } public class Cougar extends Feline { public Cougar() { System.out.print("cougar "); } public static void main(String[] args) { new Cougar().go(); } void go() { type = "c "; System.out.print(this.type + super.type); } } What is the result? A) cougar c c B) cougar c f C) feline cougar c c D) feline cougar c f E) Compilation fails F) An exception is thrown at run time.

5. Given: class Alpha { String getType() { return "alpha"; } } class Beta extends Alpha { String getType() { return "beta"; } } class Gamma extends Beta { String getType() { return "gamma"; } public static void main(String[] args) { Gamma g1 = new Alpha(); Gamma g2 = new Beta(); System.out.println(g1.getType() + " " + g2.getType()); } } What is the result? A) alpha beta B) beta beta C) gamma gamma D) alpha alpha E) Compilation fails.

6. Choose the False statement among the following? Java support globally accessible variables Java is programming language from Sun JavaScript is a programming language from Netscape None of the Above

7. The 16 bit coding scheme employed in Java programming is EDCDIC UNICODE ASCII Hexadecimal

8. What is the output of the following program? `public class example { public static void main(String args[]) { int x=0, y=2; do { x=++x; y--; } while(y>0); System.out.println(x); } }` 0 1 2 Compilation Error

9. Command to execute a compiled java programs is run execute `javac java`

10. Choose the correct statement from the given choice? Arrays in Java are essentially objects It is not possible to assign one array to another. Individual elements of array can however be assigned. Array elements are indexed from 1 to size of array If a method tries to access an array element beyond its range, a compile warning is generated.

11. Consider the following code segment. `List list = new ArrayList(); list.add("P"); list.add("Q"); list.add("R"); list.set(2, "s"); list.add(2, "T"); list.add("u"); System.out.println(list);` What is printed as a result of executing the code segment? (a) [P, Q, R, s, T] (b) [P, Q, s, T, u] (c) [P, Q, T, s, u] (d) [P, T, Q, s, u] (e) [P, T, s, R, u]

12. Consider the following instance variable and method. `private List nums; / ** Precondition: nums.size > 0 */ public void numQuest() { int k = 0; Integer zero = new Integer(0); while (k < nums.size()) { if (nums.get(k).equals(zero)) nums.remove(k); k++; } }` Assume that List nums initially contains the following Integer values. [0, 0, 4, 2, 5, 0, 3, 0] What will List nums contain as a result of executing numQuest ? (a) [0, 0, 4, 2, 5, 0, 3, 0] (b) [4, 2, 5, 3] (c) [0, 0, 0, 4, 2, 5, 3] (d) [3, 5, 2, 4, 0, 0, 0, 0] (e) [0, 4, 2, 5, 3]

13. At a certain high school students receive letter grades based on the following scale. Numeric Score Letter Grade 93 or above A From 84 to 92 inclusive B From 75 to 83 inclusive C Below 75 F Which of the following code segments will assign the correct string to grade for a given integer score ? I. `if (score >= 93) grade = "A"; if (score >= 84 && score <= 92) grade = "B"; if (score >= 75 && score <= 83) grade = "C"; if (score < 75) grade = "F";` II. `if (score >= 93) grade = "A"; if (84 <= score <= 92) grade = "B"; if (75 <= score <= 83) grade = "C"; if (score < 75) grade = "F";` III. `if (score >= 93) grade = "A"; else if (score >= 84) grade = "B"; else if (score >= 75) grade = "C"; else grade = "F";` (a) II only (b) III only (c) I and II only (d) I and III only (e) I, II, and III

14. Consider the following declarations. `public interface Comparable { int compareTo(Object other); }` `public class SomeClass implements Comparable { // ... other methods not shown }` Which of the following method signatures of compareTo will satisfy the Comparable interface requirement? I. `public int compareTo(Object other)` II. `public int compareTo(SomeClass other)` III. `public boolean compareTo(Object other)` (a) I only (b) II only (c) III only (d) I and II only (e) I, II, and III

15. Consider the following instance variable and method. `private int[] myStuff; /** Precondition: myStuff contains int values in no particular order. */ public int mystery(int num) { for (int k = myStuff.length - 1; k >= 0; k--) { if (myStuff[k] < num) { return k; } } return -1; }` Which of the following best describes the contents of myStuff after the following statement has been executed? `int m = mystery(n);` (a) All values in positions 0 through m are less than n. (b) All values in positions m+1 through myStuff.length-1 are less than n. (c) All values in positions m+1 through myStuff.length-1 are greater than or equal to n. (d) The smallest value is at position m. (e) The largest value that is smaller than n is at position m.

16. Consider the following method. `/** Precondition: x >= 0 */ public void mystery(int x) { System.out.print(x % 10); if ((x / 10) != 0) { mystery(x / 10); } System.out.print(x % 10); }` Which of the following is printed as a result of the call `mystery(1234)` ? (a) 1441 (b) 3443 (c) 12344321 (d) 43211234 (e) Many digits are printed due to infinite recursion.

17. Consider the following two classes. `public class Dog { public void act() { System.out.print("run"); eat(); } public void eat() { System.out.print("eat"); } }` `public class UnderDog extends Dog { public void act() { super.act(); System.out.print("sleep"); } public void eat() { super.eat(); System.out.print("bark"); } }` Assume that the following declaration appears in a client program. `Dog fido = new UnderDog();` What is printed as a result of the call `fido.act()` ? (a) run eat (b) run eat sleep (c) run eat sleep bark (d) run eat bark sleep (e) Nothing is printed due to infinite recursion.

18. Consider the following recursive method. `public static int mystery(int n) { if (n == 0) return 1; else return 3 * mystery(n - 1); }` What value is returned as a result of the call `mystery(5)` ? (a) 0 (b) 3 (c) 81 (d) 243 (e) 6561

19. Consider the following instance variable and method. `private int[] arr; /** Precondition: arr.length > 0 */ public int checkArray() { int loc = 5 * arr.length / 2; for (int k = 0; k < arr.length; k++) { if (arr[k] > arr[loc]) loc = k; } return loc; }` Which of the following is the best postcondition for `checkArray` ? (a) Returns the index of the first element in array `arr` whose value is greater than `arr[loc]` (b) Returns the index of the last element in array `arr` whose value is greater than `arr[loc]` (c) Returns the largest value in array `arr` (d) Returns the index of the largest value in array `arr` (e) Returns the index of the largest value in the second half of array `arr`

20. Assume the following declarations have been made. `private String s; private int n; public void changer(String x, int y) { x = x + "peace"; y = y * 2; }` Assume `s` has the value "world" and `n` is 6. What are the values of `s` and `n` after the call `changer(s, n)` ? (a) `s` is world, `n` is 6 (b) `s` is worldpeace, `n` is 6 (c) `s` is world, `n` is 12 (d) `s` is worldpeace, `n` is 12 (e) `s` is peace `n` is 12

Краткие методические указания Промежуточный тест проводится в электронной форме во время последнего в учебном периоде практического занятия. Тест состоит из 20 тестовых заданий. На выполнение теста отводится 20 минут. Во время проведения теста использование литературы и других информационных ресурсов допускается только по предварительному согласованию с преподавателем. Шкала оценки Оценка Баллы Описание 5 19–20 Процент правильных ответов от 95% до 100% 4 16–18 Процент правильных ответов от 80 до 94% 3 13–15 Процент правильных ответов от 65 до 79% 2 9–12 Процент правильных ответов от 45 до 64%

2.3.5 Типовые задания

Задача 1: Опишите класс с потоковой функцией, выводящей на экран ID потока, а затем с новой строки – текстовую строку, переданную в конструктор класса (она будет храниться как поле данных класса, строка должна быть достаточно большой, например, представлять собой отрывок из известного классического или современного текста). Напишите программу, которая создаёт несколько потоков (5 - 10). Каждый из созданных потоков выводит свой текст. При этом вывод текста синхронизирован (следующий текст не выводится, пока не закончится вывод предыдущего). Этого можно добиться, например, пометив метод, выводящий текст на консоль как `synchronized`.

Задача 2: Написать программу со следующей функциональностью: Главный поток программы должен считывать вводимые пользователем строки и помещать их в конец списка `ArrayList`. Строки длиннее 80 символов можно разрезать на несколько строк. Если пользователь вводит пустую строку – программа останавливается и выводит на консоль текущее состояние списка. Дочерний поток пробуждается каждые пять секунд и сортирует получившийся список строк в лексикографическом порядке (используйте любую сортировку). Все операции над списком должны синхронизоваться при помощи `synchronized`.

Краткие методические указания На выполнение одной практической работы отводится не более одного двухчасового занятия (не включая затраты времени на проведение промежуточного теста на последнем в учебном периоде практическом занятии). После выполнения каждой практической работы студент должен представить отчет о ее выполнении, а также, по указаниям преподавателя, выполнить дополнительные практические задания по теме работы.

Шкала оценки Оценка Баллы

Оценка	Баллы	Описание
5	73–80	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практические

		задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности
4	61–72	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации
3	49–60	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации
2	33–48	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.

2.3.6 Задания для контрольной работы

2.3.6.1 Задание к теоретической части контрольной работы

Вариант 1. Архитектурный анализ GUI-фреймворков

Разработать приложение на основе TCP-соединения, позволяющее осуществлять взаимодействие клиента и сервера по совместному решению задач обработки информации. Приложение должно располагать возможностью передачи и модифицирования получаемых (передаваемых) данных.

1. Разработать приложение – калькулятор для совершения простейших арифметических операций. Исходные параметры и тип операции (+, –, /, ×) вводятся на клиентской части и передаются серверу. Сервер возвращает клиенту результат операции.

2. Разработать приложение – чат. На сервере и клиентской части отображаются передаваемые сообщения и время их отправления.

3. Разработать приложение – генератор случайных чисел. На клиентской части вводится целое положительное число N и передается серверу, а тот в свою очередь возвращает клиенту массив случайных чисел от 1 до N.

4. Разработать приложение – поисковик слов. На сервере хранится определенный текст. На клиентской части вводится слово для поиска и передается серверу, а тот в свою очередь осуществляет поиск этого слова в тексте и возвращает клиенту все предложения, в которых встречается это слово.

5. Разработать приложение – счетчик букв. На клиентской части вводится строка и передается серверу, а тот в свою очередь осуществляет подсчет гласных и согласных букв и возвращает этот результат клиенту.

6. Разработать приложение – определитель матрицы. На клиентской части вводится исходная матрица произвольного порядка и передается серверу, а тот в свою очередь вычисляет определитель этой матрицы и возвращает результат клиенту.

7. Разработать приложение для нахождения обратной матрицы размером 3×3. Исходная матрица вводится на клиентской части и передается серверу, а тот в свою очередь возвращает клиенту обратную матрицу.

8. Разработать приложение для определения победителя лотереи. На сервере хранятся номера билетов. На каждом билете имеются 10 случайных чисел от 1 до 100. На клиентской части вводятся 10 чисел от 1 до 100, и сервер должен определить номер билета, в котором имеется больше всего совпадений с введенными числами.

9. Разработать приложение для определения призовых мест на соревнования по прыжкам в длину. На сервере хранятся фамилии участников соревнований, их идентификационные номера. На клиентской части вводятся результаты прыжков по каждому идентификационному номеру, а сервер возвращает фамилии спортсменов, занявших первое, второе и третье места.

10. Разработать приложение для определения суммы подоходного налога. На клиентской части вводятся заработные платы сотрудников предприятия и передаются серверу, а тот в свою очередь возвращает суммы налога. Причем для 41 заработной платы меньше 100 000 руб. применяется ставка налога 5 %, для заработной платы от 100 000 руб. до 500 000 руб. – ставка 10 %, для заработной платы больше 500 000 руб. – ставка 15 %.

2.3.6.2 Задание к практической части контрольной работы

Вариант 1. Разработка модуля авторизации

Используя интерфейс JDBC, совместно с СУБД MySQL создать приложение согласно варианту:

1. Приложение «Справочник телефонных номеров».
2. Приложение для учета товаров на складе.
3. Приложение для учета билетов в кинотеатре.
4. Приложение для учета билетов автовокзала.
5. Приложение учета заявок по ремонту бытовой техники.
6. Приложение учета заказов в интернет-магазине.
7. Приложение для учета успеваемости студентов.
8. Приложение «Электронный отдел кадров».
9. Приложение «Расписание движения поездов».
10. Приложение для учета оказываемых предприятием услуг.
11. Приложение для учета книг в библиотеке.
12. Приложение для учета курсов валют.
13. Приложение для учета футбольных матчей.
14. Приложение для учета выигрышных лотерейных билетов.
15. Приложение для учета результатов тестирования.

2.3.7 Методика проведения лекции-дискуссии

Цель: Формирование критического мышления при выборе архитектурных решений для GUI-приложений; развитие навыков аргументации профессиональной позиции.

Задачи:

- Анализ преимуществ и недостатков различных подходов к проектированию интерфейсов
- Поиск оптимальных решений для конкретных задач
- Развитие умения отстаивать свою точку зрения с опорой на технические аргументы

Методика проведения:

1. Назначение секретаря лекции-дискуссии и инструктаж по фиксации ключевых аргументов
2. Представление спорной темы (например: «Встроенные средства стилизации vs CSS-подобные подходы в GUI»)
3. Предварительное голосование участников по позициям
4. Свободное обсуждение с обязательным аргументированием каждой позиции
5. Промежуточное резюмирование преподавателем ключевых аргументов «за» и «против»
6. Повторное голосование и анализ изменений позиций участников

7. Формулирование выводов и практических рекомендаций

Критерии оценки:

- Секретарь лекции-дискуссии – 0,5–1,5 балла
- Активное участие с аргументированными выступлениями – 0,5–1 балл за выступление

2.3.8 Методика проведения интерактивного решения задач

Цель: Развитие навыков совместного решения профессиональных задач в условиях, приближенных к реальной разработке ПО.

Задачи:

- Формирование навыков командной работы при разработке интерфейсов
- Развитие умения находить компромиссные решения при проектировании
- Отработка методик декомпозиции сложных задач на подзадачи

Методика проведения:

1. *Подготовительный этап:* Преподаватель разрабатывает проект решения задачи с начальной архитектурой и выделенными проблемными зонами.
2. *Организационный этап:*
 - Объявление темы задачи (например: «Оптимизация производительности таблицы с 10 000 записей»)
 - Назначение секретаря и тайм-кепера
 - Распределение ролей в группе (архитектор, разработчик, тестировщик)
3. *Интеллектуальный этап:*
 - Анализ исходного кода и выявление узких мест
 - Мозговой штурм возможных решений
 - Выбор оптимального подхода и распределение задач
 - Совместная реализация решения с поочередным написанием кода
4. *Заключительный этап:*
 - Демонстрация результата и сравнение производительности до/после оптимизации
 - Обсуждение принятых решений и альтернативных подходов
 - Оценка работы группы по критериям: эффективность решения, качество кода, командная работа

Критерии оценки:

- Секретарь – 1 балл
- Активное участие в решении задачи – 1–3 балла
- Качество предложенного решения – до 2 баллов

3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков по дисциплине

3.1 Балльно-рейтинговая система оценки успеваемости по дисциплине

В целях оперативного контроля уровня усвоения материала учебной дисциплины и стимулирования активной учебной деятельности студентов (очной формы обучения) используется балльно-рейтинговая система оценки успеваемости. В соответствии с этой системой оценки студенту в ходе изучения дисциплины предоставляется возможность набрать не менее 60 баллов за текущую работу в семестре и до 40 баллов на зачете (итоговый контроль). Результирующая оценка по дисциплине складывается из суммы баллов текущего и итогового контроля.

К зачету допускаются студенты, посетившие все лекции и практические занятия, прошедшие промежуточную аттестацию, с рейтингом не менее 30 баллов.

Балльно-рейтинговая система по дисциплине «Визуальное программирование на языке Java»

Виды контроля	Максимальная сумма баллов на выполнение одного вида задания	Всего возможных за семестр работ	Итого максимальная сумма баллов
Присутствие на лекционных и практических занятиях	0,12	60	7
Работа на практических занятиях (включая интерактивные)	1,0	36	36
СРС, в т.ч.:			17
– написание конспектов лекций	0,1	10	1
– выполнение домашних заданий по разработке приложений	0,5	12	6
– подготовка рефератов по современным технологиям GUI	0,3	5	2
– участие в код-ревью проектов одногруппников	0,2	10	2
– разработка документации к проекту	0,3	5	2
– защита итогового проекта (предварительная)	1,0	4	4
Контроль СРС (тестирование, решение кейсов)	0,8	10	8
Итого за семестр	–	–	60
Зачет (защита итогового проекта)	40,0	1	40
Итого по дисциплине	–	–	100

Примечание: Для заочной формы обучения максимальная сумма баллов за СРС увеличена до 55 баллов с соответствующей корректировкой других видов контроля.

Нормы выставления дополнительных баллов

Виды работ	Максимальная сумма баллов
Публикация исходного кода проекта в открытых репозиториях (GitHub/GitLab) с документацией	3
Участие в хакатонах/конкурсах по разработке приложений	4

Доклад на студенческой конференции по тематике дисциплины	3
Итого	10

Шкала итоговой оценки

Количество баллов	Оценка
90–100	отлично
75–89	хорошо
60–74	удовлетворительно
0–59	не зачтено