Министерство науки и высшего образования Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования

**«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(МИ ВлГУ)**

**Кафедра *ИС***

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
19.05.2026

# РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

*Сравнительный анализ языков программирования*

**Направление подготовки** *09.03.02 Информационные системы и технологии*

**Профиль подготовки** *Информационные системы и технологии*

| Семестр | Трудоемкость, час./зач. ед. | Лекции, час. | Практические занятия, час. | Лабораторные работы, час. | Консультация, час. | Контроль, час. | Всего (контактная работа), час. | СРС, час. | Форма промежуточного контроля (экз., зач., зач. с оц.) |
|---|---|---|---|---|---|---|---|---|---|
| **3** | **108 / 3** | **16** | **28** | | **1,6** | **0,25** | **45,85** | **62,15** | **Зач. с оц.** |
| **Итого** | **108 / 3** | **16** | **28** | | **1,6** | **0,25** | **45,85** | **62,15** | |

**Муром, 2026 г.**

## 1. Цель освоения дисциплины

Целью освоения дисциплины (модуля) "Сравнительный анализ языков программирования" является обучение студентов методологии сравнения и оценки различных языков программирования.

Задачи дисциплины:

- знакомство с историей развития языков программирования и основными тенденциями их развития;
- изучение методов сравнения и оценки языков программирования;
- знакомство с синтаксисом и семантикой языков программирования Си, Паскаль, Java и другими;
- получение практических навыков решения задач с применением различных языковых конструкций.

## 2. Место дисциплины в структуре ОПОП ВО

Дисциплина относится к части, формируемой участниками образовательных отношений и обеспечивает понимание особенностей, преимуществ и недостатков различных языков и стилей программирования. Курс базируется на знаниях, полученных студентами в процессе изучения основных дисциплин: "Информатика" и "Технологии программирования". Углубление и расширение вопросов, изложенных в данном курсе, будет осуществляться во время работы студентов над дисциплинами: "Управление данными", "Объектно-ориентированное программирование".

## 3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

| Формируемые компетенции (код, содержание компетенции) | Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции | | Наименование оценочного средства |
|---|---|---|---|
| | Индикатор достижения компетенции | Результаты обучения по дисциплине | |
| ПК-1 Способность проводить исследования на всех этапах жизненного цикла программных средств | ПК-1.1 Применяет основные подходы исследования на всех этапах жизненного цикла программых средств | Знать основные подходы исследования на всех этапах жизненного цикла программных средств (ПК-1.1)<br>Уметь применять основные подходы исследования на всех этапах жизненного цикла программных средств (ПК-1.1)<br>Владеть навыками исследования на всех этапах жизненного цикла программных средств (ПК-1.1) | вопросы устного опроса,вопросы устного опроса, практические задания |

## 4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часов.

### 4.1. Форма обучения: очная

Уровень базового образования: среднее общее.
Срок обучения 4г.

### 4.1.1. Структура дисциплины

| № п\п | Раздел (тема) дисциплины | Семестр | Контактная работа обучающихся с педагогическим работником | | | | | | | Самостоятельная работа | Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Лекции | Практические занятия | Лабораторные работы | Контрольные работы | КП / КР | Консультация | Контроль | | |
| 1 | Введение в языки программирования | 3 | 4 | | | | | | | 62 | устный опрос |
| 2 | Базовые навыки работы с C++, Python, Java, Go | 3 | 10 | 28 | | | | | | | устный опрос, практические работы |
| 3 | Сравнительный анализ C++, Python, Java, Go | 3 | 2 | | | | | | | 0,15 | устный опрос |
| Всего за семестр | | 108 | 16 | 28 | | | | 1,6 | 0,25 | 62,15 | Зач. с оц. |
| Итого | | 108 | 16 | 28 | | | | 1,6 | 0,25 | 62,15 | |

### 4.1.2. Содержание дисциплины

#### 4.1.2.1. Перечень лекций

**Семестр 3**

*Раздел 1. Введение в языки программирования*

**Лекция 1.**

Введение в языки программирования. Классификация языков программирования (2 часа).

**Лекция 2.**

Языки, ориентированные на данные. Объектно-ориентированные языки (2 часа).

*Раздел 2. Базовые навыки работы с C++, Python, Java, Go*

**Лекция 3.**

Ввод/вывод, ветвления и циклы в разных языках программирования (2 часа).

**Лекция 4.**

Базовые операции в разных языках программирования (2 часа).

**Лекция 5.**

Организация функций и работа со строками в разных языках программирования (2 часа).

**Лекция 6.**

Структуры и классы в разных языках программирования (2 часа).

**Лекция 7.**

Работа с файлами в разных языках программирования (2 часа).

*Раздел 3. Сравнительный анализ C++, Python, Java, Go*

**Лекция 8.**

Сравнение функционала и особенностей языков C++, Python, Java, Go (2 часа).

### 4.1.2.2. Перечень практических занятий

**Семестр 3**

*Раздел 2. Базовые навыки работы с C++, Python, Java, Go*

**Практическое занятие 1**

Ввод/вывод, арифметические операции, ветвления, циклы в C++ (2 часа).

**Практическое занятие 2**

Ввод/вывод, арифметические операции, ветвления, циклы в Python (2 часа).

**Практическое занятие 3**

Ввод/вывод, арифметические операции, ветвления, циклы в Java (2 часа).

**Практическое занятие 4**

Ввод/вывод, арифметические операции, ветвления, циклы в Go (2 часа).

**Практическое занятие 5**

Функции и строки в C++ (2 часа).

**Практическое занятие 6**

Функции и строки в Python (2 часа).

**Практическое занятие 7**

Функции и строки в Java (2 часа).

**Практическое занятие 8**

Функции и строки в Go (2 часа).

**Практическое занятие 9**

Структуры и классы в C++ (2 часа).

**Практическое занятие 10**

Структуры и классы в Python (2 часа).

**Практическое занятие 11**

Структуры и классы в Java (2 часа).

**Практическое занятие 12**

Структуры и классы в Go (2 часа).

**Практическое занятие 13**

Работа с файлами в C++, Python (2 часа).

**Практическое занятие 14**

Работа с файлами в Java, Go (2 часа).

### 4.1.2.3. Перечень лабораторных работ

Не планируется.

### 4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Типы перечисления.
2. Массивы и контроль соответствия типов.
3. Операторы goto.
4. Организация подпрограмм в различных языках программирования.
5. Параметры функций и их передача.

6. Стековая архитектура.
7. Распределение памяти, динамическое распределение памяти.
8. Представление вещественных чисел.
9. Полиморфизм языков программирования.
10. Динамическая диспетчеризация.
11. Работа с базами данных в Java.
12. Структура байт-кода Java.
13. Исключения в языке PL/1. Исключения в Ada. Обработка ошибок в языке Eiffel.
14. Параллельное программирование. Что такое параллелизм.
15. Взаимные исключения.
16. Мониторы и защищенные переменные.
17. Передача сообщений.
18. Языки параллельного программирования Occam, Linda.
19. Декомпозиция программ. Раздельная компиляция. Пакеты, модули.
20. Динамический полиморфизм в языке C++.
21. Логическое программирование. Чистое логическое программирование. Унификация.
22. Язык Prolog.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

### 4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

### 4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

## 4.2 Форма обучения: заочная

Уровень базового образования: среднее профессиональное.
Срок обучения 3г 6м.

| Семестр | Трудоемкость, час./ зач. ед. | Лекции, час. | Практические занятия, час. | Лабораторные работы, час. | Консультация, час. | Контроль,час. | Всего (контактная работа), час. | СРС, час. | Форма промежуточного контроля (экз., зач., зач. с оц.) |
|---|---|---|---|---|---|---|---|---|---|
| **4** | **108 / 3** | **4** | **6** | | **2** | **0,5** | **12,5** | **91,75** | **Зач. с оц.(3,75)** |
| **Итого** | **108 / 3** | **4** | **6** | | **2** | **0,5** | **12,5** | **91,75** | **3,75** |

## 4.2.1. Структура дисциплины

| № п\п | Раздел (тема) дисциплины | Семестр | Контактная работа обучающихся с педагогическим работником | | | | | | | Самостоятельная работа | Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Лекции | Практические занятия | Лабораторные работы | Контрольные работы | КП / КР | Консультация | Контроль | | |
| 1 | Введение в языки программирования | 4 | 4 | | | | | | | 88 | устный опрос |
| 2 | Базовые навыки работы с C++, Python, Java, Go | 4 | | 6 | | | | | | 0 | устный опрос, практические работы |
| 3 | Сравнительный анализ C++, Python, Java, Go | 4 | | | | | | | | 3,75 | устный опрос |
| Всего за семестр | | 108 | 4 | 6 | | + | | 2 | 0,5 | 91,75 | Зач. с оц.(3,75) |
| Итого | | 108 | 4 | 6 | | | | 2 | 0,5 | 91,75 | 3,75 |

## 4.2.2. Содержание дисциплины

### 4.2.2.1. Перечень лекций

**Семестр 4**

*Раздел 1. Введение в языки программирования*

**Лекция 1.**

Введение в языки программирования. Процедурные/непроцедурные языки программирования (2 часа).

**Лекция 2.**

Языки, ориентированные на данные. Объектно-ориентированные языки. Логическое программирование (2 часа).

### 4.2.2.2. Перечень практических занятий

**Семестр 4**

*Раздел 2. Базовые навыки работы с C++, Python, Java, Go*

**Практическое занятие 1.**

Ввод/вывод, арифметические операции, ветвления, циклы в C++, Python, Java, Go (2 часа).

**Практическое занятие 2.**

Функции и строки в C++, Python, Java, Go (2 часа).

**Практическое занятие 3.**

Структуры и классы в C++, Python, Java, Go (2 часа).

### 4.2.2.3. Перечень лабораторных работ

Не планируется.

### 4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Типы перечисления.
2. Массивы и контроль соответствия типов.
3. Операторы goto.
4. Организация подпрограмм в различных языках программирования.
5. Параметры функций и их передача.
6. Стековая архитектура.
7. Распределение памяти, динамическое распределение памяти.
8. Представление вещественных чисел.
9. Полиморфизм языков программирования.
10. Динамическая диспетчеризация.
11. Работа с базами данных в Java.
12. Структура байт-кода Java.
13. Исключения в языке PL/1. Исключения в Ada. Обработка ошибок в языке Eiffel.
14. Параллельное программирование. Что такое параллелизм.
15. Взаимные исключения.
16. Мониторы и защищенные переменные.
17. Передача сообщений.
18. Языки параллельного программирования Occam, Linda.
19. Декомпозиция программ. Раздельная компиляция. Пакеты, модули.
20. Динамический полиморфизм в языке C++.
21. Логическое программирование. Чистое логическое программирование. Унификация.
22. Язык Prolog.
23. Сравнительный анализ C++, Python, Java, Go.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

### 4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

1. Разработать на любых двух языках следующую программу: Умножение матриц.
2. Разработать на любых двух языках следующую программу: Сортировка элементов вектора методом пузырька.
3. Разработать на любых двух языках следующую программу: Сортировка элементов вектора методом выбора.

4. Разработать на любых двух языках следующую программу: Объединение двух матриц в одну матрицу.

5. Разработать на любых двух языках следующую программу: Разбиение матрицы на две матрицы.

6. Разработать на любых двух языках следующую программу: Разбиение вектора на два вектора.

7. Разработать на любых двух языках следующую программу: Объединение двух векторов в вектор.

8. Разработать на любых двух языках следующую программу: Объединение двух векторов в матрицу, состоящую из двух столбцов.

9. Разработать на любых двух языках следующую программу: Объединение двух векторов в матрицу, состоящую из двух строк.

10. Разработать на любых двух языках следующую программу: Разбиение вектора на два вектора.

11. Разработать на любых двух языках следующую программу: Поиск позиции заданного элемента в матрице.

12. Разработать на любых двух языках следующую программу: Поиск минимального элемента и его индексов в матрице.

13. Разработать на любых двух языках следующую программу: Разложение матрицы в вектор.

14. Разработать на любых двух языках следующую программу: Составление матрицы из вектора.

15. Разработать на любых двух языках следующую программу: Удаление заданного элемента в векторе.

16. Разработать на любых двух языках следующую программу: Вставка элемента в заданную позицию вектора.

17. Разработать на любых двух языках следующую программу: Формирования вектора частот появления элементов в матрице.

18. Разработать на любых двух языках следующую программу: Поиск максимального элемента и его индексов в матрице.

19. Разработать на любых двух языках следующую программу: Вставка меньшей матрицы в большую матрицу на заданную позицию.

20. Разработать на любых двух языках следующую программу: Вставка меньшего вектора в больший вектор на заданную позицию.

21. Разработать на любых двух языках следующую программу: Поиск заданной последовательности (подвектора) в векторе.

22. Разработать на любых двух языках следующую программу: Проверка присутствия заданного элемента в векторе.

23. Разработать на любых двух языках следующую программу: Проверка присутствия заданного элемента в матрице.

24. Разработать на любых двух языках следующую программу: Выборка подматрицы из большей матрицы.

25. Разработать на любых двух языках следующую программу: Выборка подвектора из большего вектора.

26. Разработать на любых двух языках следующую программу: Определение наибольшего общего делителя двух чисел.

27. Разработать на любых двух языках следующую программу: Свертка двух векторов разного размера.

28. Разработать на любых двух языках следующую программу: Свертка двух матриц разного размера.

29. Разработать на любых двух языках следующую программу: Поиск матрицы в большей матрице.

30. Разработать на любых двух языках следующую программу: Выборка главной и побочной диагоналей из квадратной матрицы.

31. Разработать на любых двух языках следующую программу: Поворот матрицы на 90 градусов по часовой стрелке.

32. Разработать на любых двух языках следующую программу: Поворот матрицы на 90 градусов против часовой стрелки.

33. Разработать на любых двух языках следующую программу: Вычисление среднего значения элементов матрицы.

34. Разработать на любых двух языках следующую программу: Вычисление количества элементов в матрице, которые больше заданного числа.

### 4.2.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

## 5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении практических работ применяется имитационный или симуляционный подход, когда преподавателем разбирается на конкретном примере проблемная ситуация, все шаги решения задачи студентам демонстрируются при помощи мультимедийной техники. Затем студенты самостоятельно решают аналогичные задания.

## 6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

## 7. Учебно-методическое и информационное обеспечение дисциплины.

### 7.1. Основная учебно-методическая литература по дисциплине

1. Кауфман В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс]/ Кауфман В.Ш.— Электрон. текстовые данные.— Саратов: Профобразование, 2019.— 464 c. - http://www.iprbookshop.ru/88014.html

2. Липпман С. Язык программирования C++. Полное руководство [Электронный ресурс]/ Липпман С., Лажойе Ж.— Электрон. текстовые данные.— Саратов: Профобразование, 2019.— 1104 c. - http://www.iprbookshop.ru/89862.html

3. Сузи Р.А. Язык программирования Python [Электронный ресурс]: учебное пособие/ Сузи Р.А.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.— 350 c. - http://www.iprbookshop.ru/97589.html

### 7.2. Дополнительная учебно-методическая литература по дисциплине

1. Васюткина И.А. Технология разработки объектно-ориентированных программ на JAVA [Электронный ресурс]: учебно-методическое пособие/ Васюткина И.А.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2012.— 152 c. - http://www.iprbookshop.ru/45047

2. Кауфман В.Ш. Языки программирования. Концепции и принципы — М.: ДМК Пресс, 2013. - 464 c. - http://www.iprbookshop.ru/6932

3. Молдованова О.В. Языки программирования и методы трансляции [Электронный ресурс]: учебное пособие/ Молдованова О.В.— Электрон. текстовые данные.— Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2012.— 134 c. - http://www.iprbookshop.ru/54809

### 7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая

## перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

электронно-библиотечная система "IPRBooks" (http://www.iprbookshop.ru).

Программное обеспечение:

Не предусмотрено.

## 7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru

iprbookshop.ru).

mivlgu.ru/iop

## 8. Материально-техническое обеспечение дисциплины

Лаборатория информатики и программирования

12 персональных компьютеров; проектор Sanyo PDG-DSU20; экран настенный Drapper Apex Star.

## 9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины. Занятия проводятся в компьютерном классе, используя специальное программное обеспечение. Обучающиеся выполняют индивидуальную задачу в соответствии с заданием на практическую работу. Полученные результаты сводятся в отчет и защищаются по традиционной методике на следующем практическом занятии.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – зачет с оценкой. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.02 Информационные системы и технологии* и профилю подготовки *Информационные системы и технологии*
Рабочую программу составил *ст. преподаватель Булаев А.В.*_____

Программа рассмотрена и одобрена на заседании кафедры *ИС*

протокол № 21 от 05.05.2026 года.
Заведующий кафедрой *ИС* _____*Андрианов Д.Е.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 12.05.2026 года.
Председатель комиссии ФИТР _____ *Кутарова Е.И.*
(Подпись) (Ф.И.О.)

*Приложение*

**Фонд оценочных материалов (средств) по дисциплине**
Сравнительный анализ языков программирования

# 1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Вопросы для устного опроса

1. Понятие языка программирования.
2. Процедурные языки программирования.
3. Языки, ориентированные на данные.
4. Объектно-ориентированные языки. Основные понятия ООП.
5. Директивы препроцессора в C++ #include и #define.
6. Директивы условной компиляции в C++.
7. Исключения в C++. Генерация и перехват.
8. Исключения в C++. Генерация внутри функций.
9. Исключения в C++. Повторная генерация исключения.
10. Исключения в C++. Стандартные и собственные классы исключений.
11. ООП в C++. Основные принципы.
12. ООП в C++. Объявление класса.
13. ООП в C++. Объекты.
14. ООП в C++. Конструкторы.
15. ООП в C++. Деструкторы.
16. ООП в C++. Структуры.
17. ООП в C++. Объединения.
18. Непроцедурные языки.
19. Java. Основные особенности.
20. Java. Базовые типы данных.
21. Java. Основные отличия от C++.
22. Java. Объектная модель.
23. Java. Нейтральность к архитектуре.
24. Java. Встроенная безопасность.
25. Java. Многопотоковость.

**Общее распределение баллов текущего контроля по видам учебных работ для студентов**

| | | |
|---|---|---|
| Рейтинг-контроль 1 | Устный опрос (2 вопроса) | 20 |
| Рейтинг-контроль 2 | Защита практических работ | 20 |
| Рейтинг-контроль 3 | Защита практических работ | 20 |
| Посещение занятий студентом | Отметка в журнале посещений | 10 |
| Дополнительные баллы (бонусы) | | 10 |
| Выполнение семестрового плана самостоятельной работы | Защита практических работ | 20 |

# 2. Промежуточная аттестация по дисциплине

**Перечень вопросов к экзамену / зачету / зачету с оценкой.**
**Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)**

ПК-1

Блок "Знать"

Понятие языка программирования

Процедурные языки программирования

Языки, ориентированные на данные

Объектно-ориентированные языки

Директивы препроцессора в C++. include, define.

Директивы препроцессора в C++. Условная компиляция.

Исключения в C++. Генерация и перехват.

Исключения в C++. Генерация внутри функции. Повторная генерация.

Исключения в C++. Стандартные и собственные классы исключений.

Объектно-ориентированное программирование в C++. Основные принципы ООП. Классы.

Объектно-ориентированное программирование в C++. Объекты. Конструкторы. Деструкторы.

Объектно-ориентированное программирование в C++. Структуры. Объединения.

Непроцедурные языки.

Особенности языка Erlang.

Свойства Java.

Базовые типы в Java.

Основные отличия Java от C++.

Объектная модель Java.

Переносимость Java на различные архитектуры.

Безопасность в Java.

Блок "Уметь"

ПК-1

Построить систему классов для описания плоских геометрических фигур: круг, квадрат, прямоугольник. Предусмотреть методы для создания объектов, перемещения на плоскости.

Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса.

Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

Составить описание класса для объектов-векторов, задаваемых координата-ми концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления длины вектора.

Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров.

Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, вывода на экран элемента массива по заданному индексу и всего массива

Составить описание класса, обеспечивающего представление матрицы про-извольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы.

Описать класс «Домашняя библиотека», частью которого является класс «Книга». Предусмотреть возможность работы с произвольным числом книг, добавления книг в библиотеку, удаления книг из нее, вывода информации о книгах на экран. Использовать конструктор и методы класса.

Описать класс «Дата» – день, месяц (число), год. Предусмотреть возмож-ность задания месяца в виде строки, возможность вывода даты на экран, а также возможно-сти прибавления к дате указанного количества дней и вычитания их из нее.

Описать класс «Студенческая группа», частью которого является класс «Студент». Предусмотреть возможность работы с переменным числом студентов, поиска студента по фамилии, добавления и удаления записей, вывода информации о студентах на экран.

Описать класс «Строка», позволяющий выполнять основные операции: за-дание строки произвольной длины, поиск в строке подстроки, объединение двух строк, воз-вращение длины строки, вывод строки на экран.

Описать класс «Экзаменационная ведомость», частью которого является класс «Студент», хранящий информацию о студенте и его оценке на экзамене. Предусмот-реть возможность задания произвольного количества студентов, сортировки студентов по фамилии, вычисления среднего балла за экзамен, а также вывода на экран количества раз-личных оценок (неудовлетворительно, удовлетворительно, хорошо, отлично).

Описать класс «Файл», содержащий сведения о произвольном файле на диске. Реализовать методы, позволяющие получать отдельные компоненты полного пути файла: каталог файла, имя файла, расширение файла. Реализовать методы получения размера файла, удаления файла.

Описать класс «Список строк», позволяющий хранить произвольное коли-чество строк, связанных в список. Предусмотреть методы добавления и удаления строк из списка, поиска нужной строки по подстроке, вывода списка строк на экран. Использовать конструктор класса.

Описать класс «Автобусный рейс», позволяющий хранить сведения о начальной и конечной станциях, а также о свободных и занятых местах (включая их номера) Предусмотреть методы покупки и возврата билета на определённое место, получения количества свободных мест. Использовать конструктор класса.

Блок "Владеть"

ПК-1

Разработать на C++ собственные классы исключений с описанием ошибки в виде строки и массива строк.

Разработать на C++ межплатформенную программу работы с файлом (открытие, чтение, запись, поиск в файле, закрытие).

Разработать на C++ межплатформенный класс взаимного исключения.

Разработать на C++ межплатформенный класс для работы с потоками (запуск, остановка, ожидание завершения).

Разработать на Java программу, подключающуюся к БД и отображающую содержание какой-либо таблицы

**Методические материалы, характеризующие процедуры оценивания**

На основе перечня вопросов к тестированию программным комплексом информационно-образовательного портала МИ ВлГУ формируются в автоматическом режиме тестовые задания для студентов: 8 вопросов из блока 1, 4 вопроса из блока 2 и 3 вопроса из блока 3. Программный комплекс формирует индивидуальные задания для каждого зарегистрированного в системе студента и устанавливает время прохождения тестирования. Результатом тестирования является балл, рассчитанный на основе количества правильных ответов. С учетом индивидуального семестрового рейтинга студента формируется итоговый балл по курсу.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

| Оценка в баллах | Оценка по шкале | Обоснование | ***Уровень сформированности компетенций*** |
|---|---|---|---|
| Более 80 | «Отлично» | Содержание курса освоено полностью, без пробелов, необходимые практические навыки | ***Высокий уровень*** |

| | | | |
|---|---|---|---|
| | | работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному | |
| 66-80 | «Хорошо» | Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками | ***Продвинутый уровень*** |
| 50-65 | «Удовлетворительно» | Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки | ***Пороговый уровень*** |
| Менее 50 | «Неудовлетворительно» | Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки | ***Компетенции не сформированы*** |

## 3. Задания в тестовой форме по дисциплине

Примеры заданий:

1. К какому классу можно отнести подавляющее большинство скриптовых языков программирования?

- компилируемые
- *интерпретируемые
- декларативные
- интеллектуальные

2. Что объединяет языки PHP, Perl, Ruby, JavaScript и Python?

- это языки искусственного интеллекта
- *это интерпретируемые языки программирования
- это компилируемые языки программирования
- это узкоспециализированные языки программирования
- это специальные языки для реализации web-технологий

3. Какой из перечисленных языков НЕ используется для реализации фронтенд части web-приложений?

- HTML.
- CSS.
- JavaScript.
- *PHP
- TypeScript.

4. Какой из перечисленных языков транслируется в чистый JavaScript?
- VBScript
- *TypeScript
- Prolog
- Fortran
- Basic

5. Какой из перечисленных языков описания данных наиболее компактный?
- XML
- YAML
- *JSON
- Эффективность сжатия данных в перечисленных форматах одинакова.

6. Какой из перечисленных языков НЕ использует теги с угловыми скобками?
- HTML
- XML
- CSS
- RSS

7. Какое из перечисленных языков программирования используют статическую типизацию?
- *Java, C++, Haskell;
- PHP, Python, Ruby
- JavaScript, VBScript, TypeScript

8. В каких языках реализован принцип ленивых вычислений: запуск функции на выполнение нужно отложить до тех пор, пока не понадобится результат ее работы?
- в императивных
- в процедурных
- в объектно-ориентированных
- в декларативных
- *в функциональных
- в логических

9. Какой парадигмы программирования НЕ существует?
- императивное
- процедурное
- объектно-ориентированное
- * метаболическое
- декларативное
- функциональное

10. На каком преимущественно языке программирования написано ядро Линукс?
- Паскаль
- GO
- Python
- Ruby
- *Си

- R

11. Какой язык программирования напрямую выполняемый процессором?
- *машинный
- ассемблер
- java
- байт-код

12. Как называется механизм обеспечения повторного использования кода в языках с поддержкой только одиночного наследования, таких как PHP?
- Скоуп
- Рефактор
- *Трейт
- Майнинг

13. Какое свойство языка программирования отражает степень автоматического обнаружения ошибок, которое может быть выполнено транслятором или операционной средой, в которой работает программа?
- понятность
- *надежность
- гибкость
- простота
- естественность
- мобильность
- стоимость

14. Какое свойство языка программирования декларирует, что язык должен содержать такие структуры данных, управляющие структуры и операции, а также иметь такой синтаксис, которые позволяли бы отражать в программе логические структуры, лежащие в основе реализуемого алгоритма?
- понятность
- надежность
- гибкость
- простота
- *естественность
- мобильность
- стоимость

15. Какое свойство языка позволяет переносить программы, написанные на нем, с одной платформы на другую с относительной легкостью?
- понятность
- надежность
- гибкость
- простота
- естественность
- *мобильность
- стоимость

16. Как называется последовательность символов, определяющая вычисления?
Программа

17. Как называется набор правил, определяющих, какие последовательности символов составляют программу и какое вычисление описывает программа?
Язык программирования

18. Как называется наиболее популярный язык программирования, относящийся к поколению 2GL (языки среднего уровня)?

Ассемблер

19. К какому поколению относятся языки визуального программирования? Напишите номер

4

20 Какое программирование еще называют непроцедурным?

Декларативное

21. Как в языке программирования называются синтаксические возможности, применение которых не влияет на поведение программы, но делает использование языка более удобным для человека?

Синтаксический сахар

22. Как называется один из операторов синтаксического сахара, который работает также как и базовый оператор ветвления, но при его использовании применяется сокращенная форма записи?

(пример синтаксиса: $var = condition ? exp1 : exp2;)

Тернарный

23. Напишите оператор объединения с null для языка PHP, являющийся синтаксическим сахаром для достаточно распространённого действия, когда совместно используются тернарный оператор и функция isset(). Он возвращает первый операнд, если он задан и не равен null, а в обратном случае возвращает второй операнд.

??

24. Какой язык программирования широко применяется для анализа данных и машинного обучения, в том числе благодаря библиотекам Numpy и Pandas?

Python

25. Какой язык программирования, используемый для программирования серверной части web-приложений, имеет следующие варианты расшифровки: "препроцессор гипертекста", "личная страница"?

PHP

26. Название какого декларативного языка программирования, основанного на формально-логической модели представления знаний, можно расшифровать как "Программирование логики"?

Пролог

27. Как называется функция логического типа, часто используемая в декларативных языках на основе формально-логической модели представления знаний?

Предикат

28. Напишите название компилируемого многопоточного языка программирования, разработанного внутри компании Google

Go

29. Как называется технология увеличения производительности программных систем, использующих байт-код, путём компиляции байт-кода в машинный код или в другой формат

непосредственно во время работы программы? Буквальный перевод названия технологии: "компиляция точно в нужное время". Напишите аббревиатуру технологии.

JIT

30. Как называется формальное описание языков программирования, которое позволяет математически интерпретировать «смысл» компьютерной программы, написанной на этом языке?

Семантика

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/mod/quiz/view.php?id=59077

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.