

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н.Г. ЧЕРНЫШЕВСКОГО»

Факультет компьютерных наук и информационных технологий

УТВЕРЖДАЮ
Декан факультета компьютерных наук
и информационных технологий
информатика / С.В. Миронов
"13" сентября 2021г.

Рабочая программа дисциплины

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

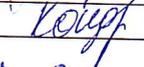
Направление подготовки бакалавриата
44.03.01 – Педагогическое образование

Профиль подготовки бакалавриата
Информатика

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Очная

Саратов,
2021

Статус	ФИО	Подпись	Дата
Преподаватель-разработчик	Портенко М.С.		13.09.21
Председатель НМК	Кондратова Ю.Н.		13.09.21
Заведующий кафедрой	Огнева М.В.		13.09.21
Специалист Учебного управления			

1. Цели освоения дисциплины

Целями освоения дисциплины «Языки программирования высокого уровня» являются:

- Понимание назначения языков программирования высокого уровня;
- Получение навыка работы с языками высокого уровня на примере языков C#, Java, C++;
- Понижение порога вхождения в изучение языков программирования высокого уровня;
- Умение находить общее и различия в реализации принципов объектно-ориентированного программирования в языках высокого уровня на примере языков C#, Java, C++;
- Получение навыка демонстрации полученных результатов исследования сравнения языков программирования;
- Получение навыка выбора языка программирования, учитывая особенности, и реализации решения задачи.

2. Место дисциплины в структуре ООП

Данная учебная дисциплина Б1.В.ДВ.03.02 «Языки программирования высокого уровня» относится к части Блока 1 «Дисциплины (Модули) и рассред. практики» учебного плана ООП и направлена на формирование у обучающихся универсальных и профессиональных компетенций.

Для изучения дисциплины необходимы компетенции, сформированные в ходе изучения Б1.В.01 «Программирование», Б1.В.03 «Теоретические основы информатики», Б1.В.06 «Структуры данных и алгоритмы».

Компетенции, сформированные при изучении данной дисциплины, используются при изучении дисциплин Б1.О.21 «Подготовка школьников к единому государственному экзамену», Б1.О.22 «Подготовка школьников к олимпиадами и конкурсам по информатике и программированию», Б1.О.12 «Теория и методика обучения информатике».

3. Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора (индикаторов) достижения компетенции	Результаты обучения
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.	1.1_Б.УК-1. Анализирует задачу, выделяя ее базовые составляющие. Осуществляет декомпозицию задачи. 2.1_Б.УК-1. Находит и критически анализирует информацию, необходимую	Знать: -основные синтаксические конструкции языков высокого уровня (например, C#, Java, C++) Уметь: -решать алгоритмические задачи на языках (например, C#, Java, C++), -сравнивать возможности языков высокого уровня и

	<p>для решения поставленной задачи.</p> <p>3.1_ Б.УК-1. Рассматривает различные варианты решения задачи, оценивая их достоинства и недостатки.</p> <p>4.1_ Б.УК-1. Грамотно, логично, аргументированно формирует собственные суждения и оценки.</p> <p>Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.</p> <p>5.1_ Б.УК-1. Определяет и оценивает практические последствия возможных решений задачи.</p>	<p>выбирать необходимый язык программирования в зависимости от задачи,</p> <p>Владеть:</p> <ul style="list-style-type: none"> -навыком применения особенностей языков программирования для анализа и синтеза информационных систем и процессов; -навыком обработки ошибок времени выполнения через работу с исключениями.
<p>ПК - 7. Способен использовать математический аппарат, методы программирования и современные информационно-коммуникационные технологии для решения практических задач получения, хранения, обработки и передачи информации</p>	<p>ПК - 7.1. Решает практические задачи получения, хранения, обработки и передачи информации.</p> <p>ПК - 7.2. Использует математический аппарат, методы программирования и современные информационно-коммуникационные технологии для решения учебных задач.</p>	<p>Знать</p> <ul style="list-style-type: none"> -механизмы передачи данных в разных языках программирования между компонентами системы; <p>Уметь</p> <ul style="list-style-type: none"> -обрабатывать данные, выполнять запросы над данными используя средства языков программирования (C#, Java, C++); <p>Владеть</p> <ul style="list-style-type: none"> -навыками использования тренажеров, онлайн конструкторов для решения задач.

4. Структура и содержание дисциплины «Языки программирования высокого уровня»

Общая трудоемкость дисциплины составляет 8 зачетных единиц 288 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					КС Р	Формы текущего контроля успеваемости (<i>по неделям семестра</i>) Формы промежуточной аттестации (<i>по семестрам</i>)
				Лек ции	Практические занятия		Общая трудое мкость	Из них – лаб. пр. подг отов ка		
ла б	пр ак т									
1	Платформа .Net. Язык С#	5	1-2	4	4	4	1	4	Практические задачи	
2	Основные операторы С#	5	3-4	4	4	4	1	2	Практические задачи	
3	Работа со строками С#	5	5-6	4	4	4	1	4	Практические задачи	
4	Работа с коллекциями С#	5	7-8	4	4	4	1	4	Практические задачи	
5	Работа с файлами С#	5	9	2	2	2	1	2	Практические задачи	
6	Объектно-ориентированное программирование С#	5	10- 11	4	4	4	2	7	Практические задачи	
7	LINQ	5	12- 13	4	4	4	1	7	Практические задачи Контрольная работа	
Промежуточная аттестация 36									Экзамен, контрольная работа	
Итого			144	26	26	26	8	30		
6 семестр										
8	Платформа Java. Язык Java	6	1	1	1	1	1	2	Практические задачи	
9	Основные операторы Java	6	1	1	1	1	0	1	Практические задачи	
10	Работа со строками Java	6	2	1	1	1	1	2	Практические задачи	

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					КС Р	Формы текущего контроля успеваемости (<i>по неделям семестра</i>) Формы промежуточной аттестации (<i>по семестрам</i>)
				Лек ции	Практические занятия		Общая трудое мкость	Из них – лаб. пр. подг отов ка		
ла б	пр ак т	КС Р								
11	Работа с коллекциями Java	6	2	1	1	1	1	2	Практические задачи	
12	Работа с файлами Java	6	3	1	1	1	0	1	Практические задачи	
13	Объектно-ориентированное программирование Java	6	3-4	3	3	3	1	4	Практические задачи, контрольная работа	
14	Stream	6	5-6	4	4	4	1	4	Практические задачи	
15	Язык C++, сравнение общих характеристик с C#, Java	6	7	2	2	2	0	2	Практические задачи	
16	Основные операторы C++, сравнение с C#, Java	6	8	2	2	2	0	2	Практические задачи	
17	Работа со строками C++, сравнение с C#, Java	6	9	2	2	2	0	2	Практические задачи	
18	Работа с коллекциями C++, сравнение с C#, Java	6	10	2	2	2	1	2	Практические задачи	
19	Работа с файлами C++, сравнение с C#, Java	6	11	2	2	2	0	0	Практические задачи	
20	Объектно-ориентированное программирование C++, сравнение с C#, Java	6	12	2	2	2	1	3	Практические задачи	
21	Библиотека algorithm	6	13	2	2	2	1	3	Практические задачи	
	Промежуточная аттестация 36								Экзамен, контрольная работа	
	Итого		144	26	52		8	30		

5 семестр

Платформа .Net. Язык C#

История создания. Компоненты платформы и их назначение. Принцип компилирования C#. Система типов. Сборка мусора, управление памятью, указатели. Работа с исключительными ситуациями.

Основные операторы C#

Операторы присваивания, ветвления, цикла, операции: сравнения, побитовые, арифметические, логические. Работа с классом DateTime.

Работа со строками C#

Типы работы со строками String и String Builder. Основное назначение, фокусы внимания, нюансы использования. Regex – работа с регулярными выражениями.

Работа с коллекциями C#

Стандартные коллекции C#, назначение, нюансы использования, иерархия, стандартные интерфейсы коллекций.

Работа с файлами C#

Классы File, FileInfo, Directory, DirectoryInfo, FileStream, StreamReader, StreamWriter, BinaryReader, BinaryWriter.

Объектно-ориентированное программирование C#

Реализация основных принципов ООП в C# (наследование, инкапсуляция, полиморфизм). Понятие интерфейса. Интерфейс IComparable. Делегаты, события, анонимные типы, анонимные методы, лямбда-выражения.

LINQ

Язык запросов LINQ. Назначение, формат данных для обработки. Форматы XML, JSON. Проекция данных, фильтрация коллекций, сортировка, объединение, пересечение, разность коллекций, агрегатные операции, группировка, проверка наличия, отложенное и немедленное выполнение, делегаты в запросах LINQ.

6 семестр

Платформа Java. Язык Java

История создания. Компоненты платформы и их назначение. Принцип компилирования Java. Система типов. Сборка мусора, управление памятью, указатели. Сравнение принципов работы с C#.

Основные операторы Java

Операторы присваивания, ветвления, цикла, операции: сравнения, побитовые, арифметические, логические. Сравнение с C#.

Работа со строками Java

Типы по работе со строками. Сравнение с C#

Работа с коллекциями Java

Работа с основными коллекциями в Java. Иерархия коллекций и интерфейсов коллекций в Java. Сравнение с C#.

Работа с файлами Java

Принципы работы с файлами в Java, сравнение с C#.

Объектно-ориентированное программирование Java

Реализация основных принципов ООП в Java (наследование, инкапсуляция, полиморфизм). Понятие интерфейса. Сравнение с C#.

Stream

Назначение, функционал, возможности Stream. Сравнение с LINQ от C#.

Язык C++, сравнение общих характеристик с C#, Java

Особенности языка C++, назначение. Работа с памятью.

Основные операторы C++, сравнение с C#, Java

Операторы присваивания, ветвления, цикла, операции: сравнения, побитовые, арифметические, логические. Сравнение с C#, Java.

Работа со строками C++, сравнение с C#, Java

Типы по работе со строками в C, C++. Сравнение с C#, Java.

Работа с коллекциями C++, сравнение с C#, Java

Работа с основными коллекциями C++, сравнение с C# и Java.

Работа с файлами C++, сравнение с C#, Java

Основные способы работы с файлами, сравнение с C# и Java.

Объектно-ориентированное программирование C++, сравнение с C#, Java

Реализация принципов ООП в C++, проблема множественного наследования, сравнение с C# и Java.

Библиотека `algorithm`

Функциональность библиотеки.

5. Образовательные технологии, применяемые при освоении дисциплины «Языки программирования высокого уровня»

При преподавании дисциплины «Языки программирования высокого уровня» предусматривается широкое использование в учебном процессе активных и интерактивных форм проведения занятий.

В рамках практических занятий по данной дисциплине используются кейс-задания, выполнение которых направлено на формирование таких профессиональных действий как выявление применимости того или иного языка для решения конкретной задачи, нахождения компромиссов, определение назначения, устранения влияния заблуждений и критического отношения к источникам информации. Предусмотрено время на подготовку, проведение и обсуждение выступлений студентов по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской темы. В ходе работы используются тренажеры, например, контестер – система автоматической проверки задач на school.sgu.ru. тренажер на w3schools.com.

Для организации работы студентов в рамках данной дисциплины предусмотрена поддержка в рамках курса на платформе системы управления учебным процессом LMS Moodle (school.sgu.ru). Система позволяет выставлять задание, на которое студент может ответить в виде файла с аннотацией, возможна организация коллективной работы через выкладывание материалов выступлений на форуме, организован выбор доступных тем для выступлений и т.д. Система также имеет функционал для людей с ограниченными возможностями здоровья.

При обучении лиц с ограниченными возможностями здоровья и инвалидов используются подходы, способствующие созданию безбарьерной среды: технологии дифференциации и индивидуализации обучения, увеличение времени на самостоятельное освоение материала.

Удельный вес занятий, проводимых в интерактивных формах, составляет 50% аудиторных занятий.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Возможные темы выступлений:

1. Использование сортировок для элементов массива (аналогичных коллекций) в языке C#/ Java/ C++.
2. Сравнение работы с регулярными выражениями в C# и Java.
3. Проблема множественного наследования, варианты решений в разных языках.
4. Работа LINQ с XML форматом C#.
5. Использование предикатов в C++.
6. Понятие интерфейса в C#.
7. Сборка мусора в C#.

8. Возможные варианты организации сравнения объектов в Java.

Примеры заданий для контрольной работы 1. 5 семестр

Вариант 1.

Для класса

```
public class Student  
{  
    public string First { get; set; }  
    public string Last { get; set; }  
    public int ID { get; set; }  
    public List<int> Scores;  
}
```

создайте список студентов из не менее 20 студентов с не менее 10 оценками по сто балльной системе в Scores. Сделайте возможным выбор заполнения списка из двух вариантов – через файл и через заполнение списка инициализаторами объектов. Для данного списка получите результаты двумя способами: по запросу и через методы расширения, используя лямбда выражения.

1. выборка всех студентов, у которых первая оценка превышает порог в 70 минимальных баллов;
2. выборка всех студентов, у которых первая оценка находится в диапазоне 70-80 баллов;
3. выборка всех студентов, у которых все оценки превышают 80 баллов.
4. упорядочите полученные результаты
 - a. по ID по возрастанию;
 - b. по имени и фамилии по убыванию;
5. Примените группировку к исходному списку студентов по баллам за последний экзамен, не забудьте посмотреть результат выполнения запроса;
6. Упорядочите группы по знаку их ключа;
7. Создайте по исходному списку только список имен, сначала выведя все имена, включая повторения, затем каждое имя только один раз, упорядочите полученный результат по возрастанию.

Примеры заданий Контрольной работы 2. 6 семестр

Реализация класса в Java, демонстрация работы с ним.

Для каждого класса необходимо:

1. Реализовать три конструктора:
 - a. конструктор без параметров,
 - b. конструктор с параметрами,
 - c. конструктор-копию;
2. Реализовать не менее трех методов (допускается создание private вспомогательных методов)

- a. вывода данных на экран,
- b. методы в контексте конкретного класса;
3. Реализовать сравнение элементов класса.
4. Продемонстрировать работу с данным классом:
 - a. создать коллекцию (массив) экземпляров данного класса (предпочтительно из файла),
 - b. отсортировать стандартным методом (для C++ метод `sort()` из библиотеки `<algorithm>`).

Если это будет необходимо, допустимо создание вспомогательных классов, связанных с данным в задании. Например, класс Дата (для работы с классом Продукт), или класс Инструктор (для работы с классом Дисциплина), или класс

Адрес.

Варианты

- Класс Person

Данные: фамилия, имя, отчество, городской телефон, мобильный телефон

Методы:

1. возвращающий город по коду городского телефона
2. возвращающий название мобильного оператора по префиксу мобильного телефона

Операция сравнения: лексикографический порядок по фамилии, имени, отчеству

- Класс Vehicle

Данные: фирма-изготовитель, модель, расход топлива на 100км, цена

Методы:

1. изменения цены
2. получения названия модели

Операция сравнения: по расходу топлива на 100км

- Класс Product

Данные: фирма-изготовитель, название, цена, дата производства, срок годности

Методы:

1. проверки истечения срока годности
2. изменения цены

Операция сравнения: продолжительность срока годности

- Класс Course

Данные: название курса, инструктор, количество часов

Методы:

1. получения названия курса
2. изменения количества часов

Операция сравнения: по названию курса в лексикографическом порядке

Контрольные вопросы по освоению дисциплины «Языки программирования высокого уровня»

1. История создания платформы .Net.
2. Назначение CLR.
3. Назначение CTS.
4. Назначение и принципы работы сборщика мусора.
5. Принцип компилирования C#.
6. Ошибки времени выполнения. Работа с исключительными ситуациями.
7. Система типов: типы значения и ссылочные типы.
8. Работа с указателями в C#.
9. Операторы ветвления C# If.
10. Операторы ветвления C# case.
11. Операции: C# сравнения,
12. Операции C# побитовые,
13. Операции C# арифметические,
14. Операции C# логические.
15. Операторы цикла в C#: for.
16. Операторы цикла в C#: while.
17. Операторы цикла в C#: do while.
18. Работа с массивами в C#.
19. Операторы цикла в C#: foreach.
20. Типы работы со строками в C#: String.
21. Типы работы со строками в C#:
22. Выбор между String и String Builder в C#.
23. Regex – работа с регулярными выражениями.
24. Еще классы по работе с регулярными выражениями.
25. Работа с массивами в C#. Массив как объект.
26. Интерфейс ICollection в C#.
27. Интерфейс IList в C#.
28. Обобщенные типы в C#.
29. Работа с Dictionary<K,V> в C#.
30. Работа со Stack <T> в C#
31. Работа с Queue<T> в C#
32. Стандартные интерфейсы коллекций в иерархии интерфейсов.
33. Классы File, FileInfo, работа с ними в C#.
34. Классы Directory, DirectoryInfo, работа с ними в C#.
35. Работа с FileStream в C#.
36. Работа с StreamReader, StreamWriter в C#,
37. Работа с BinaryReader, BinaryWriter C#.
38. Объектно-ориентированное программирование C#. Класс object.
39. Реализация основных принципов ООП в C# (наследование, инкапсуляция, полиморфизм).
40. Перегрузка операций в C#.
41. Понятие интерфейса в C#.

- 42.Интерфейс IComparable и его назначение.
- 43.Делегат в C#.
- 44.Событие в C#.
- 45.Анонимный тип в C#.
- 46.Анонимный метод в C#.
- 47.Взаимосвязь делегатов, событий, анонимных типов, анонимных методов, лямбда-выражений в C#.
- 48.Язык запросов LINQ. Назначение, формат данных для обработки.
- 49.Формат XML.
- 50.Формат JSON.
- 51.Проекция данных в LINQ C#.
- 52.Фильтрация коллекций в LINQ C#.
- 53.Сортировка в LINQ C#.
- 54.Объединение коллекций в LINQ C#.
- 55.Пересечение коллекций в LINQ C#.
- 56.Разность коллекций в LINQ C#.
- 57.Агрегатные операции в LINQ C#.
- 58.Группировка данных в LINQ C#.
- 59.Проверка наличия записи в данных в LINQ C#.
- 60.Отложенное и немедленное выполнение в LINQ C#.
- 61.Делегаты в запросах LINQ C#..
- 62.Похожа ли компиляция программы в Java на C# (или наоборот).
- 63.Есть ли аналог делегата в C++.
- 64.Есть ли аналог делегата в Java.
- 65.Есть ли цикл foreach в C++.
- 66.Есть ли цикл foreach в Java.
- 67.Есть ли коллекции формата <ключ, значение> в C++.
- 68.Есть ли коллекции формата <ключ, значение> в Java.
- 69.Есть ли обобщенные типы в C++. Или аналог.
- 70.Есть ли обобщенные типы в Java. Или аналог.
- 71.Есть ли работа аналогичная работе с исключительными ситуациями в C# в Java и C++.
- 72.Есть ли класс object с похожей функциональностью на C# в Java и C++.
- 73.Есть ли понятие интерфейса в C++, в Java.
- 74.История создания Java. Компоненты платформы и их назначение. Сравнение с C#
- 75.Принцип компилирования Java.
- 76.Принцип компилирования Java. Сравнение с C#.
- 77.Система типов Java: primitive, non-primitive.
78. Сборка мусора, управление памятью, указатели в Java.
- 79.Сборка мусора, управление памятью, указатели в Java. Сравнение принципов работы с C#.
- 80.Основные операторы Java. Условный, цикла. Сравнение с C#.
- 81.Операции: сравнения, побитовые, арифметические, логические. Сравнение с C#.

82. Перегрузка операций в Java. Сравнение с C#.
83. Работа со строками Java.
84. Работа со строками Java. Сравнение с C#.
85. Типы по работе со строками отличные от C#.
86. Работа с коллекциями Java.
87. Иерархия коллекций и интерфейсов коллекций в Java. Сравнение с C#.
88. Работа с файлами Java
89. Принципы работы с файлами в Java, сравнение с C#.
90. Объектно-ориентированное программирование Java.
91. Класс object в Java.
92. Класс object в Java, сравнение с C#.
93. Реализация основных принципов ООП в Java (наследование, инкапсуляция, полиморфизм). Понятие интерфейса. Сравнение с C#.
94. Stream в Java. Сравнение с C#.
95. Особенности языка C++, назначение. Работа с памятью.
96. Язык C++, сравнение общих характеристик с C#, Java.
97. Операторы присваивания, ветвления, цикла, операции: сравнения, побитовые, арифметические, логические. Сравнение с C#, Java.
98. Работа со строками C++.
99. Работа со строками C++, сравнение с C#, Java.
100. Работа с коллекциями STL C++, сравнение с C#, Java
101. Работа с массивами C++, сравнение с C# и Java.
102. Шаблоны методов в C++.
103. Работа с файлами C++, сравнение с C#, Java
104. Основные способы работы с файлами, сравнение с C# и Java.
105. Объектно-ориентированное программирование C++.
106. Объектно-ориентированное программирование C++, сравнение с C#, Java.
107. Перегрузка операций сравнения в C++, аналоги в C#, Java.
108. Проблема множественного наследования C++, сравнение с C# и Java.
109. Библиотека algorithm.
110. Примеры использования функциональности библиотеки algorithm.
111. Что такое предикат в C++.
112. Примеры использования предикатов.

Вопросы к экзамену. 5 семестр

1. История создания платформы .Net.
2. Компоненты платформы .Net и их назначение.
3. Принцип компилирования C#.
4. Система типов.
5. Ошибки времени выполнения. Работа с исключительными ситуациями в C#.
6. Сборка мусора, управление памятью, указатели.

7. Операторы ветвления C#
8. Операции: C# сравнения, побитовые, арифметические, логические.
9. Операторы цикла в C#.
10. Типы работы со строками String и String Builder. Основное назначение, фокусы внимания, нюансы использования.
11. Regex – работа с регулярными выражениями.
12. Работа с массивами в C#
13. Интерфейс IEnumerable
14. Работа с Dictionary<K,V> в C#
15. Работа со Stack <T> в C#
16. Работа с Queue<T> в C#
17. Стандартные интерфейсы коллекций.
18. Классы File, FileInfo, работа с ними в C#.
19. Классы Directory, DirectoryInfo, работа с ними в C#.
20. Работа с FileStream в C#.
21. Работа с StreamReader, StreamWriter в C#,
22. Работа с BinaryReader, BinaryWriter C#.
23. Объектно-ориентированное программирование C#.
24. Класс object и его роль.
25. Реализация основных принципов ООП в C# (наследование, инкапсуляция, полиморфизм).
26. Понятие интерфейса в C#.
27. Понятия и взаимосвязь делегатов, событий, анонимных типов, анонимных методов, лямбда-выражений в C#.
28. Язык запросов LINQ в C#
29. Язык запросов LINQ. Назначение, формат данных для обработки.
30. Форматы XML, JSON.
31. Команды LINQ .
32. Отложенное и немедленное выполнение запросов LINQ
33. Делегаты в запросах LINQ.

6 семестр

1. Платформа Java. Язык Java. Назначение.
2. Платформа Java. Язык Java. Назначение. Сравнение с C#.
3. История создания. Компоненты платформы и их назначение.
4. История создания. Компоненты платформы и их назначение. Сравнение с C#
5. Принцип компилирования Java.
6. Принцип компилирования Java. Сравнение с C#.
7. Система типов Java: primitive, non-primitive.
8. Сборка мусора, управление памятью, указатели в Java.
9. Сборка мусора, управление памятью, указатели в Java. Сравнение принципов работы с C#.
10. Основные операторы Java. Условный, цикла. Сравнение с C#.

11. Операции: сравнения, побитовые, арифметические, логические.
Сравнение с C#.
12. Перегрузка операций в Java. Сравнение с C#.
13. Работа со строками Java.
14. Работа со строками Java. Сравнение с C#.
15. Типы по работе со строками отличные от C#
16. Работа с коллекциями Java.
17. Иерархия коллекций и интерфейсов коллекций в Java. Сравнение с C#.
18. Работа с файлами Java
19. Принципы работы с файлами в Java, сравнение с C#.
20. Объектно-ориентированное программирование Java.
21. Класс object в Java.
22. Класс object в Java, сравнение с C#.
23. Реализация основных принципов ООП в Java (наследование, инкапсуляция, полиморфизм). Понятие интерфейса. Сравнение с C#.
24. Stream в Java. Сравнение с C#.
25. Особенности языка C++, назначение. Работа с памятью.
26. Язык C++, сравнение общих характеристик с C#, Java.
27. Операторы присваивания, ветвления, цикла, операции: сравнения, побитовые, арифметические, логические. Сравнение с C#, Java.
28. Работа со строками C++.
29. Работа со строками C++, сравнение с C#, Java.
30. Работа с коллекциями STL C++, сравнение с C#, Java
31. Работа с массивами C++, сравнение с C# и Java.
32. Работа с файлами C++, сравнение с C#, Java
33. Основные способы работы с файлами, сравнение с C# и Java.
34. Объектно-ориентированное программирование C++.
35. Объектно-ориентированное программирование C++, сравнение с C#, Java.
36. Проблема множественного наследования, сравнение с C# и Java.
37. Библиотека algorithm. Функциональность библиотеки.
38. Работа с предикатами в C++.

7. Данные для учета успеваемости студентов в БАРС

Таблица 1.1 Таблица максимальных баллов по видам учебной деятельности.

1	2	3	4	5	6	7	8	9
Семестр	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
1	10	20	10	20	0	0	40	100
2	10	20	10	20	0	0	40	100

Программа оценивания учебной деятельности студента

5 семестр

Лекции: посещаемость, активность, выступления – 0 - 10 баллов.

Лабораторные занятия: своевременность, аккуратность решения, грамотное обоснование решения, объяснение решения – 0 – 20 баллов.

Практические занятия: уровень подготовки к занятиям, подготовка к выступлению, участие в обсуждениях – 0 – 10 баллов. Выполнение контрольной работы

Самостоятельная работа: регулярность самостоятельной работы, грамотность подготовки к занятиям, обращение с различными источниками, готовность к контрольной работе – 0 – 20.

Автоматизированное тестирование: *Не предусмотрено.*

Другие виды учебной деятельности: *Не предусмотрено.*

Промежуточная аттестация:

При определении баллов при аттестации преподаватель может воспользоваться следующим примером ранжирования:

- 25-30 баллов – ответ на «отлично»
- 20-24 баллов – ответ на «хорошо»
- 18-21 баллов – ответ на «удовлетворительно»
- 0-17 баллов – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за первый семестр по дисциплине «Языки программирования высокого уровня» составляет 100 баллов.

Таблица 2.1 Таблица пересчета полученной студентом суммы баллов по дисциплине «Языки программирования высокого уровня» (практике) в оценку (зачет):

60 баллов и более	и	«зачтено» (при недифференцированной оценке)
меньше 60 баллов		«не зачтено»

Таблица 2.2 Таблица пересчета полученной студентом суммы баллов по дисциплине «Языки программирования высокого уровня» в оценку (экзамен):

90 – 100 баллов	«отлично»
80 – 89 баллов	«хорошо»
60 – 79 баллов	«удовлетворительно»
0 – 59 баллов	«не удовлетворительно»

6 семестр

Лекции: посещаемость, активность, выступления – 0 - 10 баллов.

Лабораторные занятия: своевременность, аккуратность решения, грамотное обоснование решения, объяснение решения – 0 – 20 баллов.

Практические занятия: уровень подготовки к занятиям, подготовка к выступлению, участие в обсуждениях – 0 – 10 баллов. Выполнение контрольной работы

Самостоятельная работа: регулярность самостоятельной работы, грамотность подготовки к занятиям, обращение с различными источниками, готовность к контрольной работе – 0 – 20.

Автоматизированное тестирование: *Не предусмотрено.*

Другие виды учебной деятельности: *Не предусмотрено.*

Промежуточная аттестация:

При определении баллов при аттестации преподаватель может воспользоваться следующим примером ранжирования:

- 25-30 баллов – ответ на «отлично»
- 20-24 баллов – ответ на «хорошо»
- 18-21 баллов – ответ на «удовлетворительно»
- 0-17 баллов – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за первый семестр по дисциплине «Языки программирования высокого уровня» составляет 100 баллов.

Таблица 2.1 Таблица пересчета полученной студентом суммы баллов по дисциплине «Языки программирования высокого уровня» (практике) в оценку (зачет):

60 баллов и более	и	«зачтено» (при недифференцированной оценке)
меньше 60 баллов		«не зачтено»

Таблица 2.2 Таблица пересчета полученной студентом суммы баллов по дисциплине «Языки программирования высокого уровня» в оценку (экзамен):

90 – 100 баллов	«отлично»
80 – 89 баллов	«хорошо»
60 – 79 баллов	«удовлетворительно»
0 – 59 баллов	«не удовлетворительно»

8. Учебно-методическое и информационное обеспечение дисциплины (модуля).

а) литература:

1. Голицына, О. Л. Языки программирования : учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 399 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-613-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1209231>– Режим доступа: по подписке.
2. Дадян, Э. Г. Современные технологии программирования. Язык C# : учебник : в 2 томах. Для начинающих пользователей / Э.Г. Дадян. — Москва : ИНФРА-М, 2021. — 312 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1196552. - ISBN 978-5-16-016613-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1196552>– Режим доступа: по подписке.
3. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1020593>. – Режим доступа: по подписке.
4. Огнева, Марина Валентиновна. Программирование в среде Visual Studio.Net: разработка приложений на языке C# / М. В. Огнева, Е. В. Кудрина. - Саратов : КУБиК, 2010. - 545 с. - [Кудрина] . - Библиогр.: с. 545 (5 назв.). - ISBN 978-5-91818-079-2 : 138.00 р. - Текст : непосредственный.

в) программное обеспечение и Интернет-ресурсы

1. Среды разработки:
 - a. Visual Studio (C#, C++) – студенческая подписка, или аналоги
 - b. Visual Studio Code(C#, C++)
 - c. Intelij Idea (java) – студенческая подписка, или аналоги
 - d. Eclipse (java, C++)– открытая подписка
2. Stream in Java <https://www.geeksforgeeks.org/stream-in-java/?msclkid=3fde5943b62d11ec878c0045e1f844c4>
3. LINQ in C# <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/?msclkid=9a7dae1fb62d11ec963ec1b652b96793>
4. Библиотека <algorithm> в C++ <https://cplusplus.com/reference/algorithm/>
5. Учебник с встроенными упражнениями и компилятором по C++ <https://www.w3schools.com/cpp/default.asp>
6. Учебник с встроенными упражнениями и компилятором по java https://www.w3schools.com/java/java_exercises.asp
7. Учебник с встроенными упражнениями и компилятором по C# https://www.w3schools.com/cs/cs_exercises.php
8. Задачник с системой автоматической проверки задач (есть компиляторы C#, java, C++) <https://school.sgu.ru/course/view.php?id=181>
9. Регулярные выражения в Java <https://www.geeksforgeeks.org/regular-expressions-in-java/?ref=lbp>

9. Материально-техническое обеспечение дисциплины (модуля)

Для проведения лекционных занятий необходима лекционная аудитория, снабженная компьютером и проекционной системой.

Для проведения семинарских занятий необходима аудитория, снабженная досками.

Реализация *практической подготовки* в рамках учебных занятий запланирована на базе кафедры информатики и программирования, а также образовательного центра непрерывной подготовки IT-специалистов.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению 44.03.01 – Педагогическое образование, профиль подготовки «Информатика».

Автор
ст. преподаватель

М.С. Портенко

Программа одобрена на заседании кафедры информатики и программирования от 13 сентября 2021 года, протокол № 2