

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 КНиИТ,  
к. ф. м. н., доцент  
С. В. Миронов  
«31» 08 2021 г.

**Рабочая программа дисциплины**  
**ЯЗЫКИ ПРОГРАММИРОВАНИЯ**




Направление подготовки бакалавриата  
09.03.04 Программная инженерия

Профиль подготовки бакалавриата  
Разработка программно-информационных систем

Квалификация (степень) выпускника  
Бакалавр

Форма обучения  
Заочная

Саратов,  
2021 год

Статус	ФИО	Подпись	Дата
Преподаватель-разработчик	С. В. Миронов		31.08.2021
Председатель НМК	Ю. Н. Кондратова		31.08.2021
Заведующий кафедрой	С. В. Миронов		31.08.2021
Специалист Учебного управления			

## **1. Цели освоения дисциплины**

Цели освоения дисциплины «Языки программирования»— знакомство слушателя с основными концепциями современных языков программирования высокого уровня. Содержание курса включает основные сведения о характеристиках и свойствах современных языков программирования, принципы построения языков и технологии программирования, средства описания данных; средства описания действий; средства абстракции.

Задачи курса:

- познакомить студента с основными направлениями в развитии языков программирования;
- рассмотреть базовый инструментарий языков программирования;
- рассмотреть причины возникновения, использования и популярности языков программирования;
- обучить студента основам современных языков программирования.

## **2. Место дисциплины в структуре ООП**

Данная учебная дисциплина относится к обязательной части Блока 1 «Дисциплины (Модули)» ООП, является обязательной для изучения и способствует формированию общепрофессиональных компетенций.

Для изучения дисциплины необходимы базовые знания в области информатики, программирования, а также компетенции полученные в результате изучения курсов «Информационные технологии и программирование», «Теоретическая информатика».

Программа курса ориентирована на возможность расширения профессиональных навыков для успешной дальнейшей деятельности. Предполагаемое изучение в рамках курса дополнительных языков программирования направлено на повышение профессиональной мобильности слушателя.

Курс «Языки программирования» содержательно и методологически взаимосвязан с курсами «Тестирование программного обеспечения», «Технологии программирования», «Базы данных».

Компетенции, полученные слушателем в результате изучения курса, могут оказаться полезными при прохождении студентом курсов «Логическое и функциональное программирование», «Стандартизация ПО», «Интеллектуальные системы и технологии»

### 3. Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикаторов достижения компетенции	Результаты обучения
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знает — Стандартные библиотеки выбранного языка программирования
	ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знает — Методологии разработки программного обеспечения и технологии программирования Умеет — Использовать выбранную среду программирования
	ОПК-2.3. Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Знает — Методологии разработки программного обеспечения и технологии программирования Владеет навыками — Разработки программного обеспечения
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1. Знает основные алгоритмы решения типовых задач, области и способы их применения, методы и приемы алгоритмизации поставленных задач, методы повышения читаемости программного кода	Знает — Различные парадигмы программирования — Основные принципы написания читабельного программного кода Умеет — Писать программный код в соответствии с заданными правилами оформления

<p>ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов</p>	<p>ОПК-6.1. Знает основные алгоритмы решения типовых задач, области и способы их применения, методы и приемы алгоритмизации поставленных задач, методы повышения читаемости программного кода</p>	<ul style="list-style-type: none"> <li>— Выделять в данном программном коде отклонения от заданных правил оформления</li> <li>Владеет навыком</li> <li>— Разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов</li> <li>— Использования особенностей языка программирования для построения эффективных программ</li> <li>— Написания программного кода, соответствующего заданным правилам оформления</li> <li>— Написания выделения в заданном программном коде несоответствий заданным стандартам</li> </ul>
	<p>ОПК-6.2. Умеет применять выбранные языки программирования для написания программного кода, использовать современные компиляторы, отладчики и оптимизаторы программного кода, методы и приемы отладки программного кода</p>	<p>Знает</p> <ul style="list-style-type: none"> <li>— Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования</li> <li>— Особенности различных языков программирования и их влияние на эффективность разрабатываемого программного кода</li> <li>— Базовые способы верификации написанного программного кода</li> </ul> <p>Умеет</p> <ul style="list-style-type: none"> <li>— Применять выбранные языки программирования для написания программного кода</li> <li>— Применять выбранные языки программирования для реализации заданных алгоритмов</li> <li>— Оценивать правильность работы данного программного кода</li> </ul> <p>Владеет навыком</p> <ul style="list-style-type: none"> <li>— Использования особенностей языка программирования для построения эффективных программ</li> <li>— Чтения, понимания и верификации данного программного кода</li> </ul>

<p>ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов</p>	<p>ОПК-6.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Знает</p> <ul style="list-style-type: none"> <li>— Язык программирования на котором выполнен программный код</li> <li>— Особенности различных языков программирования и их влияние на эффективность разрабатываемого программного кода</li> </ul> <p>Умеет</p> <ul style="list-style-type: none"> <li>— Читать программные коды</li> </ul> <p>Владеет навыком</p> <ul style="list-style-type: none"> <li>— Чтения и написания программного кода с применением различных парадигм программирования</li> <li>— Разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов</li> </ul>
<p>ОПК-7. Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой</p>	<p>ОПК-7.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий</p>	<p>Знает</p> <ul style="list-style-type: none"> <li>— Общие сведения о лексике, синтаксисе и семантике языка программирования</li> <li>— Парадигмы программирования</li> <li>— Варианты типизации, исполнения конструкций, диспетчеризации в языках программирования</li> </ul> <p>Владеет навыком</p> <ul style="list-style-type: none"> <li>— Использования методик написания читабельного программного кода в условиях использования различных сред программирования</li> </ul>



1	2	3	4	5	6	7	8	9
1.	Общие принципы построения и использования языков программирования.	4	–	24	2	2	20	
2.	Средства описания действий в современных языках программирования.	4	–	24	2	2	20	
3.	Средства описания данных в современных языках программирования.	4	–	24	2	2	20	
	ИТОГО			72	6	6	60	
4.	Средства описания абстракций в современных языках программирования.	5	–	16	1		15	
5.	Стандартные библиотеки языков программирования.	5	–	16	1		15	Контрольная работа
6.	Функциональное программирование.	5	–	17	1	1	15	
7.	Элементы языка программирования.	5	–	12	1	1	10	
8.	Функции.	5	–	12	1	1	10	
9.	Элементы данных.	5	–	12	1	1	10	
10.	Построение абстракций.	5	–	14	2	2	10	
	Промежуточная аттестация							Экзамен
	ИТОГО			108	8	6	85	9
11.	Полиморфизм и наследование.	6	–	16		1	15	
12.	Получение результата вычислений.	6	–	8			8	
13.	Типизация.	6	–	10			10	
14.	Макросы.	6	–	17		1	16	
15.	Связывание.	6	–	11		1	10	
16.	Параметры процедур.	6	–	6			6	
17.	Объектно-ориентированное программирование.	6	–	17		1	16	
18.	Логическое программирование.	6	–	6			6	
19.	Общие вопросы языков программирования.	6	–	8			8	Контрольная работа
	Промежуточная аттестация							Экзамен
	ИТОГО			108		4	95	9
	<b>ВСЕГО</b>			<b>288</b>	<b>14</b>	<b>16</b>	<b>240</b>	<b>18</b>

#### 4.1. Содержание учебной дисциплины

ТЕМА 1. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ И ИСПОЛЬЗОВАНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. Классификация языков программирования. Императивные

(процедурные) языки программирования. Декларативные языки программирования. Функциональные языки программирования. Логические языки программирования. Объектно-ориентированные языки программирования. Языки сценариев. Языки параллельных вычислений. Подходы к программированию: структурный, модульный; функциональный; логический; объектно-ориентированный; смешанный; компонентно-ориентированный; чисто объектный. Пути развития и классификация языков программирования. Основные понятия языков программирования.

ТЕМА 2. СРЕДСТВА ОПИСАНИЯ ДЕЙСТВИЙ В СОВРЕМЕННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ. Операторы: оператор присваивания, управляющий оператор, специальные операторы. Операторы управления: операторы ветвления, циклы, переходы, блок. Составной оператор или блок.

ТЕМА 3. СРЕДСТВА ОПИСАНИЯ ДАННЫХ В СОВРЕМЕННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ. Структура ЯП. Базисные средства: скалярные (простые типы, операции), структурные (структуры и конструкции). Средства развития: поддержка библиотек, типы данных, абстрактные типы данных, модули, классы. Средства защиты: абстрактные типы данных, инкапсуляция. Простые типы данных и их виды: арифметические типы, логические типы, символьные типы. Структурные типы данных и их виды: массив, запись (объединение), множество, файл, строка. Динамические структуры данных и их виды: связанные списки, стеки, очереди, деревья, деки, кольца. Односвязные списки и их элементы.

ТЕМА 4. СРЕДСТВА ОПИСАНИЯ АБСТРАКЦИЙ В СОВРЕМЕННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ. Спецификация, реализация, параметризация, разновидности классов, классы и объекты в ООП. Сообщения, экземпляры и инициализация, синтаксис пересылки сообщений, способы создания и инициализации. Абстрактные типы данных и методология объектно-ориентированного программирования. Инкапсуляция, классы и объекты, параметризация, спецификация, реализация. Модули. Интуитивное описание наследования, формы наследования, преимущества наследования. Повторное использование кода. Композиция и наследование. Полиморфные функции, абстракции низкого и высокого уровней. Разновидности полиморфизма, полиморфные переменные, перегрузка, переопределение, отложенные методы, чистый полиморфизм спецификация, реализация, параметризация, классы и объекты. Функторы.

ТЕМА 5. СТАНДАРТНЫЕ БИБЛИОТЕКИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. Организация библиотеки языка. Средства подключения и использования библиотеки. Возможности стандартных библиотек.

ТЕМА 6. ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ. Императивные (процедурные) и декларативные языки программирования. Функциональные языки программирования. Базовые средства функциональных языков программирования. Основные признаки функционального программирования.



ТЕМА 7. ЭЛЕМЕНТЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ. Лексика, синтаксис и семантика языка программирования. Выполнение программы. Динамическое окружение, порядок его формирования и использования. Затенение. Семантика языковой конструкции.

ТЕМА 8. ФУНКЦИИ. Семантика определения и вызова функции в случае неизменяемых переменных. Рекурсивные функции. Рекурсивный и итерационный процесс. Хвостовая рекурсия. Древоподобная рекурсия. Взаимная рекурсия.

ТЕМА 9. ЭЛЕМЕНТЫ ДАННЫХ. Составные структуры данных. Кортежи и списки. Контейнеры. Их основные предназначения. Примеры контейнеров. Объекты первого класса. Функции как объекты первого класса. Анонимные функции. Их предназначения. Каррирование и частичное выполнение функции. Каррированная функция.

ТЕМА 10. ПОСТРОЕНИЕ АБСТРАКЦИЙ. Конструкторы. Геттеры и сеттеры. Барьеры абстракций. Сопоставление с образцом. Модули. Определение модулей на примере Standard ML.

ТЕМА 11. ПОЛИМОРФИЗМ И НАСЛЕДОВАНИЕ. Полиморфные типы данных. Примеры описания собственных полиморфных типов на Standard ML и Haskell. Полиморфные операции. Организация полиморфизма операций на Haskell.

ТЕМА 12. ПОЛУЧЕНИЕ РЕЗУЛЬТАТА ВЫЧИСЛЕНИЙ. Аппликативный и нормальный порядок вычислений. Мемоизация. Ленивые (отложенные) вычисления. Формы выражений в случае ленивых вычислений. Слабая головная нормальная форма. Сопоставление с образцом в случае ленивых вычислений. Эффективность хвостовой рекурсии в случае ленивых вычислений.

ТЕМА 13. ТИПИЗАЦИЯ. Явная и неявная типизация. Механизм вывода типов. Статическое окружение. Порядок его формирования и использования. Статическая и динамическая типизация. Утиная типизация. Сильная и слабая типизация. Предназначение системы типизации. Полная (complete) система типизации и надежная (sound) система типизации.

ТЕМА 14. МАКРОСЫ. Макросы. Отличие макросов от функций. Протечки в макросах. Избавление от протечек.

ТЕМА 15. СВЯЗЫВАНИЕ. Локальные и глобальные переменные. Области видимости. Лексическое и динамическое связывание. Замыкание. Порядок его формирования и использования в различных средах. Моделирование отложенных вычислений и мемоизации на Lisp.

ТЕМА 16. ПАРАМЕТРЫ ПРОЦЕДУР. Обязательные и необязательные параметры функции. Описание необязательных параметров. Передача параметров по ссылке и по значению.

ТЕМА 17. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ. Объектноориентированное и параллельное программирование. Событийно-управляемое программирование. Инкапсуляция, полиморфизм и наследование как свойства объектно-ориентированного программирования. Динамическая диспет-

черизация. Двойная диспетчеризация. Интерфейсы (примеси).

ТЕМА 18. ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ. Основные концепции логического программирования. Порядок выполнения программы на языке логического программирования. Унификация. Бэктрекинг.

ТЕМА 19. ОБЩИЕ ВОПРОСЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. История развития языков программирования высокого уровня. Экологическая ниша языка программирования. Предпосылки возникновения новых языков программирования. Свойства хорошего языка программирования. Языки численных и символьных вычислений.

#### ***4.2. План лабораторных занятий***

В рамках лабораторных занятий 4-го семестра студенты знакомятся с одним из современных языков программирования высокого уровня. Лабораторные занятия проводятся в интерактивной форме. Материал соответствующей темы занятия подается преподавателем и прорабатывается студентами в ходе разбора примеров.

Номер занятия	Тема	Содержание лабораторных занятий
1	2	3
1	Общие принципы построения и использования языков программирования.	Первое знакомство с языком программирования; варианты реализаций языка программирования; среда программирования; общие принципы взаимодействия со средой программирования; структура программы; средства взаимодействия программы и пользователя; написание и запуск первой программы; варианты использования написанных программ.
1	Средства описания данных в современных языках программирования.	Знакомство с элементарными типами данных языка программирования. Рассмотрение вариантов использования таких типов данных.
1	Средства описания действий в современных языках программирования.	Операторы: оператор присваивания, управляющий оператор. Рассмотрение вариантов использования операторов.
1	Средства описания действий в современных языках программирования.	Операторы: операторы ветвления, циклы, переходы, блок. Рассмотрение вариантов использования операторов.
1	Средства описания действий в современных языках программирования.	Специальные операторы. Рекурсивные алгоритмы. Рекурсивный и итерационный процесс.
2	Средства описания данных в современных языках программирования.	Скалярные типы данных (арифметические типы, логические типы, символьные типы). Средства для работы со скалярными типами данных. Структурные типы данных (массив, запись (объединение), множество). Средства для работы с ними.
2	Средства описания данных в современных языках программирования.	Структурные типы данных (строка, файл). Средства для работы с ними.
2	Средства описания данных в современных языках программирования.	Динамические структуры данных и их виды: связанные списки, стеки, очереди, деревья, деки, кольца. Организация динамических структур. Инкапсуляция.
2	Средства описания данных в современных языках программирования.	Функции более высокого порядка. Организация алгоритмов с использованием функций более высокого порядка.
2	Средства описания абстракций в современных языках программирования.	Средства описания абстракций. Инкапсуляция, ее назначение.
2	Средства описания абстракций в современных языках программирования.	Классы и объекты. Привязка объекта к классу. Иерархия классов.
3	Средства описания абстракций в современных языках программирования.	Наследование. Варианты наследования в языке.
3	Средства описания абстракций в современных языках программирования.	Полиморфизм. Функторы.

1	2	3
3	Стандартные библиотеки языков программирования.	Рассмотрение стандартных библиотек языка и вариантов их использования.

В рамках лабораторных занятий 5-го семестра студенты знакомятся с концепциями, изложенными на лекциях в ходе написания программ на различных языках программирования. Лабораторные занятия проводятся в интерактивной форме. Материал соответствующей темы занятия подается преподавателем и прорабатывается студентами в ходе разбора заданий лабораторных работ. В течение семестра студент должен выполнить 6 лабораторных работ.

Номер занятия	Тема	Содержание лабораторных занятий
1	2	3
1	Лабораторная работа 0. «Знакомство»	Знакомство студента со средой программирования, с механизмом сдачи/проверки решений лабораторных работ.
1	Лабораторная работа 1. «Люди – животные»	Знакомство с базовыми средствами функциональных языков программирования; написание программы в языке программирования со строгой статической явной типизацией; оперирование базовыми типами данных языка Standard ML; работа со сложными типами данных: списки, кортежи, контейнеры Option; использование рекурсивных вычислений вместо конструкций цикла.
1	Лабораторная работа №2. «Uno»	Продолжение знакомства со средствами функциональных языков программирования; написание программы в языке программирования с неявной типизацией; работа со сложными типами данных: списки, кортежи, структуры, контейнеры, пользовательские типы данных; использование функций высших порядков; сопоставление с образцом (шаблоном) при определении функций и инициализации имен; реализация итерационного процесса в виде функции с хвостовой рекурсией.
2	Лабораторная работа №3. «Бесконечные вычисления»	Оперирование бесконечными списками в языке с ленивыми вычислениями; проведение процесса вычислений с использованием бесконечных списков как стандартных интерфейсов; использование рекурсивных вычислений вместо конструкций цикла.

1	2	3
2	Лабораторная работа №4. «MUPL»	Знакомство с процессом формирования динамического окружения при выполнении программы; знакомство с механизмом замыкания; оперирование пользовательскими типами данных на языке Haskell; написание программ на языке программирования с ограниченным набором операций; первоначальное знакомство с механизмом макросов.
3	Лабораторная работа №5. «Опять производные»	Знакомство с базовыми принципами символьных вычислений; написание программы в языке программирования с динамической типизацией; знакомство с механизмом взаимной рекурсии; оперирование базовыми структурами данных языка Lisp.
3	Лабораторная работа №6. «Морской бой»	Знакомство с базовыми принципами объектно-ориентированного программирования; написание программы в объектно-ориентированном языке программирования с динамической типизацией; оперирование базовыми приемами программирования на языке Ruby.

## 5. Образовательные технологии, применяемые при освоении дисциплины

В рамках лекционных занятий проводится знакомство студентов с терминологией дисциплины, дается необходимая теоретическая информация, касающаяся базовых концепций современных языков программирования. В рамках лабораторных занятий студенты более детально знакомятся с одним из современных языков программирования. Как лекционные, так и лабораторные занятия проводятся в интерактивной форме, предполагающей активное участие студента в обсуждении вопросов дисциплины. Для подачи материала используются мультимедийные презентации.

Для закрепления навыков работы с изучаемым языком программирования студентам предлагается выполнить ряд заданий самостоятельно.

В рамках курса планируется использование средств порталов [course.sgu.ru](http://course.sgu.ru) и <http://oecdo.sgu.ru/> для интерактивного общения студентов и преподавателя в рамках самостоятельной работы и для проведения учета текущей успеваемости студентов.

К образовательному процессу планируется подключение ведущих специалистов компаний «Мирантис ИТ», «EPAM Systems».

При обучении лиц с ограниченными возможностями и инвалидов используются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуализации обучения, сопровождение тьюторами в образовательном пространстве; увеличивается время на самостоятельное освоение материала.

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

**6.1. Фонд оценочных средств**

Фонд оценочных средств дисциплины включает в себя контрольные вопросы, задания контрольных работ.

**6.2. Виды самостоятельной работы обучающихся:**

1. Изучение основной и дополнительной литературы по материалам курса.
2. Выполнение заданий самостоятельной работы по курсу. Каждому студенту предлагается набор заданий (перечислены в Фонде оценочных средств) для реализации на изучаемом языке программирования. По каждому из заданий студенту выдается отдельный номер варианта.

## 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
Се- мestr	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
4	5	10	0	0	0	0	0	15
5	5	15	0	30	0	25	10	85
ИТОГО	10	25	0	30	0	25	10	100
6	0	40	0	25	0	25	10	100

### Программа оценивания учебной деятельности студента Семестр 4

**Лекции.** Посещаемость, опрос, активность и др. — от 0 до 5 баллов.

**Лабораторные занятия.** Выполнение заданий лабораторной работы, посещаемость, опрос, активность и др. — от 0 до 10 баллов.

**Практические занятия.** Не предусмотрены.

**Самостоятельная работа.** Баллы за самостоятельную работу не предусмотрены.

**Автоматизированное тестирование.** Не предусмотрено.

**Другие виды учебной деятельности.** Не предусмотрены.

**Промежуточная аттестация.** Не предусмотрена.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за один семестр по дисциплине «Языки программирования» составляет 15 баллов.

### Семестр 5

**Лекции.** Посещаемость, опрос, активность и др. за один семестр — от 0 до 5 баллов.

**Лабораторные занятия.** Выполнение заданий лабораторной работы, посещаемость, опрос, активность и др. — от 0 до 15 баллов.

**Практические занятия.** Не предусмотрены.

**Самостоятельная работа.**

- Контроль выполнения заданий самостоятельной работы — от 0 до 20 баллов;
- Контрольная работа (от 0 до 10 баллов);

Таким образом, студент в течении одного семестра может получить от 0 до 30 баллов.

**Автоматизированное тестирование.** Не предусмотрено.

**Другие виды учебной деятельности.** Выполнение факультативных заданий, изучение факультативного материала по дополнительным главам дисциплины, своевременность выполнения текущих и дополнительных заданий — от 0 до 25 баллов.

**Промежуточная аттестация.** Проводится в форме устного экзамена.

Методика оценивания знаний обучающихся по дисциплине «Языки программирования» в ходе промежуточной аттестации (экзамена):

8–10 баллов:

Ответ студента содержит:

- глубокое знание программного материала, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой;
- знание концептуально-понятийного аппарата всего курса;
- знание монографической литературы по курсу,
- также свидетельствует о способности:
- самостоятельно критически оценивать основные положения курса;
- увязывать теорию с практикой.

6–7 баллов:

Ответ студента свидетельствует:

- о полном знании материала по программе;
- о знании рекомендованной литературы,
- а также содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

1–5 баллов:

Ответ студента содержит:

- поверхностные знания важнейших разделов программы и содержания лекционного курса;



- затруднения с использованием научно-понятийного аппарата и терминологии курса;
- стремление логически четко построить ответ, а также свидетельствует о возможности последующего обучения.

Студенту, имеющему существенные пробелы в знании основного материала по программе, а также допустившему принципиальные ошибки при изложении материала ставится оценка 0 баллов.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 5 семестр по дисциплине «Языки программирования» составляет 85 баллов.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 4-й и пятый семестр по дисциплине «Языки программирования» составляет 100 баллов.

**Таблица 2.1 Таблица пересчета полученной студентом суммы баллов по дисциплине «Языки программирования» в оценку (экзамен):**

от 85 баллов и более	«отлично»
от 65 до 84 баллов	«хорошо»
от 50 до 64 баллов	«удовлетворительно»
меньше 50 баллов	«неудовлетворительно»

## **Семестр 6**

**Лекции.** Не предусмотрены.

**Лабораторные занятия.** Выполнение заданий лабораторных работ — от 0 до 40 баллов.

**Практические занятия.** Не предусмотрены.

**Самостоятельная работа.** Контроль выполнения заданий самостоятельной работы в течение одного семестра — от 0 до 25 баллов.

**Автоматизированное тестирование.** Не предусмотрено.

**Другие виды учебной деятельности.**

- Посещаемость лабораторных занятий — от 0 до 3 баллов;
- Проверка лабораторных работ других студентов потока — от 0 до 8 баллов;
- Проверка ответов других студентов на вопросы теоретического опроса — от 0 или 1 балл;

- Выполнение факультативных заданий — от 0 до 5 баллов;
- Написание реферата — от 0 до 2 баллов;
- Изучение факультативного материала по дополнительным главам дисциплины, своевременность выполнения текущих и дополнительных заданий, дополнительные выдающиеся успехи/достижения студента не связанные с непосредственным выполнением лабораторных или факультативных заданий — от 0 до 6 баллов.

Таким образом, студент в течении одного семестра может получить от 0 до 25 баллов за другие виды учебной деятельности.

**Промежуточная аттестация.** Промежуточная аттестация состоит из двух частей. Первая часть — контрольная работа в форме тестирования студента в рамках системы moodle на портале course.sgu.ru, за которую он может получить от 0 до 3 баллов.

Вторая часть — устный экзамен, за который студенту может быть начислено от 0 до 7 баллов.

Методика оценивания знаний обучающихся по дисциплине «Языки программирования» в ходе промежуточной аттестации (экзамена):

6–7 баллов:

Ответ студента содержит:

- глубокое знание программного материала, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой;
- знание концептуально-понятийного аппарата всего курса;
- знание монографической литературы по курсу,
- также свидетельствует о способности:
- самостоятельно критически оценивать основные положения курса;
- увязывать теорию с практикой.

4–5 баллов:

Ответ студента свидетельствует:

- о полном знании материала по программе;
- о знании рекомендованной литературы,
- а также содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

1–3 баллов:

Ответ студента содержит:

- поверхностные знания важнейших разделов программы и содержания лекционного курса;
- затруднения с использованием научно-понятийного аппарата и терминологии курса;
- стремление логически четко построить ответ, а также свидетельствует о возможности последующего обучения.

Студенту, имеющему существенные пробелы в знании основного материала по программе, а также допустившему принципиальные ошибки при изложении материала ставится оценка 0 баллов.

Таким образом, в течение семестра студент может получить от 0 до 10 за промежуточную аттестацию.

Максимально возможная сумма баллов за все виды учебной деятельности студента за 6 семестр по дисциплине «Языки программирования» составляет 100 баллов.

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

от 65 баллов и более	«отлично»
от 55 до 64 баллов	«хорошо»
от 39 до 54 баллов	«удовлетворительно»
меньше 39 баллов	«неудовлетворительно»

## **8. Учебно-методическое и информационное обеспечение дисциплины «Языки программирования»**

а) литература:

1. *Голицына, О. Л., Попов, И. И., Партыка, Т. Л.* Языки программирования — Москва : Издательство «ФОРУМ», 2010. <http://znanium.com/go.php?id=226043> (Электронный ресурс)
2. *Еленев, В. Д., Гоголев, М. Ю.* Алгоритмические языки и технологии программирования на языках высокого уровня — Самар. гос. аэрокосм. ун-т им. акад. С. П. Королева (нац. исслед. ун-т). — Самара : Изд-во СГАУ, 2010. <http://rucont.ru/efd/230047> (Электронный ресурс)

в) Интернет-ресурсы:

3. *Пайн, К.* Электронный учебник по языку Ruby в переводе Шохирева М. В. <http://www.shokhirev.com/mikhail/ruby/ltp/>
4. *Роганов, Е. А., Роганова, Н. А.* Программирование на языке Ruby <http://36.msiu.ru/books/7>

г) программное обеспечение:

Свободное программное обеспечение:

5. Adobe Reader.
6. Google Chrome или аналог.
7. 7-Zip.
8. Sublime Text.
9. Standard ML of New Jersey (SML/NJ).
10. Hugs 98 (WinHugs).
11. Steel Bank Common Lisp.
12. Ruby.
13. SWI-Prolog.

## **9. Материально-техническое обеспечение учебной дисциплины «Языки программирования»**

Лекционная аудитория с мультимедийным оборудованием и с выходом в Интернет.

Компьютерный класс с оборудованием для показа мультимедийных презентаций, с возможностью работы под управлением операционной системы Linux и операционной системы Windows, с подключением к Internet, рассчитанный на обучение группы студентов из 8–12 человек, удовлетворяющий санитарно-гигиеническим требованиям.

Программа составлена в соответствии с требованиями ФГОС ВО для направления 09.03.04 «Программная инженерия» и профиля подготовки «Разработка программно-информационных систем» (квалификация (степень) «бакалавр»).

Автор: зав. кафедрой, кандидат физико-математических наук, доцент,  
С. В. Миронов

Программа разработана в 2020 г. (Одобрена на заседании кафедры математической кибернетики и компьютерных наук 23 апреля 2020 года протокол № 17).

Программа актуализирована в 2021 г. (Одобрена на заседании кафедры математической кибернетики и компьютерных наук 13.09.2021 протокол № 2).