

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ГУМАНИТАРНОЙ СФЕРЕ
*Учебно-научный центр программного и лингвистического обеспечения
интеллектуальных систем*

КОНЦЕПЦИИ И ТЕХНОЛОГИИ СОВРЕМЕННОГО ПРОГРАММИРОВАНИЯ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

45.03.04 Интеллектуальные системы в гуманитарной сфере

Разработка и программирование интеллектуальных систем в гуманитарной сфере

Уровень высшего образования: бакалавриат

Форма обучения очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2022

Концепции и технологии современного программирования
Рабочая программа дисциплины
Составитель:
доцент
М.Е. Епифанов

УТВЕРЖДЕНО
Протокол заседания кафедры МЛиИС
№ 5 от 24.03.2022

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1 Цели и задачи дисциплины

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения дисциплине

1.3. Место дисциплины в структуре образовательной программы

2. Структура дисциплины

3. Содержание дисциплины

4. Образовательные технологии

5. Оценка планируемых результатов обучения

5.1. Система оценивания

5.2. Критерии выставления оценок

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

7. Материально-техническое обеспечение дисциплины

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья

9. Методические материалы

9.1. Планы практических (семинарских, лабораторных) занятий

9.2. Методические указания для обучающихся по освоению дисциплины

Приложения

Приложение 1. Аннотация дисциплины

Приложение 2. Лист изменений

1. Пояснительная записка

1.1. Цели и задачи дисциплины

Цели дисциплины.

Основных целей курса две:

- дать студенту целостное представление о концепциях и технологиях современного программирования, тем самым, резюмируя и обобщая его знания, полученные при изучении всех других дисциплин по направлению «Программирование».
- способствовать дальнейшему формированию у студента навыков работы самостоятельного программиста.

Задачи дисциплины.

Для этого решаются следующие образовательные *задачи*:

в рамках данного курса предлагается обзор основных концепций и технологий, применяемых в языках программирования и других средствах программирования, в частности, реализуемых в современных инструментальных системах проектирования и программирования, рассматриваются тенденции дальнейшего развития выразительных средств и технологий.

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения дисциплине:

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
<p>ПК-3 Способен разрабатывать и тестировать новые программы и интерфейсы систем</p> <p>ПК-1 Способен разрабатывать алгоритмы обработки информации с использованием современных математических методов</p>	<p>ПК-3.3 Имеет практический опыт разработки и тестирования прикладных программ</p> <p>ПК-1.3 Имеет практический опыт разработки алгоритмов обработки информации с использованием современных математических методов</p>	<p><i>Знать:</i> Студенты должны овладеть следующими общими понятиями, связанными с технологиями современного программирования и развитием поддерживающих эти технологии программных средств:</p> <ul style="list-style-type: none"> ● предметная область, прикладная задача, приложение; ● модель (предметной области, приложения); ● «метафора» (совокупность идей, положенных в основу) технологии/средства; ● основанный на воплощении метафоры «подход к решению» (задачи), стиль программирования (задачи); ● реализующие некоторую метафору (см. выше) адекватные средства программирования (возможно, выразительные средства в языке программирования). <p>Студенты должны иметь представление:</p> <ul style="list-style-type: none"> ● об истории, современном состоянии и тенденциях развития основных концепций, технологий и средств (включая языки) программирования; ● об архитектуре программных (в том числе распределенных) приложений. <p><i>Уметь:</i> грамотно разрабатывать алгоритмы и их программные решения, комплексно используя применяемые для разработки адекватные инструментальные средства.</p> <p><i>Владеть</i> навыками самостоятельного специалиста в области программной реализации прикладных приложений.</p>

ПК-3 Способен разрабатывать и тестировать новые программы и интерфейсы систем	ПК-3.2 Умеет использовать интегрированные среды разработки, включая средства визуального программирования, умеет использовать средства автоматизации этапов анализа и проектирования	<p><i>Знать</i> основные принципы использования интегрированных сред разработки (IDE) программных приложений.</p> <p>Студенты должны иметь представление (на элементарном, популярном, начальном уровне) об инструментальных средствах проектирования программных приложений.</p> <p><i>Уметь</i> по возможности, адекватно, выбирать средства для разработки программных приложений, исходя из особенностей прикладной задачи и предметной области.</p> <p><i>Владеть</i> навыками: анализа документации к программным системам; программиста, умеющего разбираться в новых для себя средствах программирования.</p>
---	--	---

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Концепции и технологии современного программирования» относится к формируемой участниками образовательных отношений части блока Б1 дисциплин учебного плана.

Предполагается, что приступающий к изучению настоящей дисциплины студент успешно освоил все другие курсы для бакалавров по направлению «Программирование», и тем самым, овладел соответствующей суммой **знаний и умений**, а также **владеет** навыками, приобретенными в процессе изучения этих курсов.

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: выполнение ВКР бакалавра, а также связанные с программированием дисциплины, изучаемые в магистратуре ОИС в ГС.

2. Структура дисциплины

Структура дисциплины для очной формы обучения

Общая трудоёмкость дисциплины составляет 3 з.е., 108 ч., в том числе контактная работа обучающихся с преподавателем 42 ч., самостоятельная работа обучающихся 48 ч.

№ п/п	Раздел дисциплины/темы	Се м е ст р	Виды учебной работы (в часах)			Формы текущего контроля успеваемости, форма промежуточной
			Контактная	Про меж уточ ная	Са мо сто яте	

			Лекции	Семинары	Практические занятия	Лабораторные занятия	аттестация	ль-ная работа	аттестации (по семестрам)
1.	Развитие выразительных и инструментальных средств программирования.	8	4			2		8	Опрос, оценка выполнения практических заданий
2.	Программа = алгоритм + данные.	8	2			30		32	Опрос, оценка выполнения практических заданий
3	Пакеты решений. Обзор средств проектирования программных приложений.	8	4					8	Опрос
	Итоговая аттестация	8					18		экзамен
	Итого		10			32	18	48	

3. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Развитие выразительных и инструментальных средств программирования.	<p>Программный продукт vs «уникальная» программа. Время жизни программного продукта. Основные этапы создания и эксплуатации. «Программы и люди» – обзор различных категорий (ролей) разработчиков и пользователей. Противоречивые интересы, другие проблемы и конфликты. «Программы и программы» – проблемы взаимодействия. Экономика программирования (краткий экскурс). Важные полезные свойства программ: модифицируемость, переносимость, повторное использование кода и т.п.</p> <p>Выразительные средства декомпозиции задач в программных системах.</p> <p>Обзор основных стилей программирования (процедурное, функциональное, логическое программирование).</p> <p>Обзор парадигм и технологий программирования: историческая ретроспектива, настоящее время, тенденции развития.</p>

2.	Программа = алгоритм + данные.	<p>Программа = алгоритм + данные (тезис Н. Вирта). Отделение данных от «процедурного» кода как одна из основных тенденций программирования. Примеры соответствующих технических решений в прикладных программах. Декларативность функционального и логического программирования. Примеры элементов декларативного программирования в различных парадигмах и технологиях процедурного программирования: - интерфейсы в модульном программировании, ООП, компонентном программировании (в том числе в COM: язык IDL, автоматизация и диспетчерские интерфейсы, интерфейс IDispatch, его метод Invoke, библиотеки типа); - визуальное программирование (обзор IDE, декларативное определение элементов интерфейса разрабатываемой программы и т.п.) ...</p>
3	Пакеты решений. Обзор средств проектирования программных приложений.	<p>Объектноориентированный подход к проектированию программных приложений, подход Г.Буча. Язык UML (обзор). Некоторые другие нотации описания бизнес-процессов (краткий обзор ARIS, IDEF и его модификаций, DFD и т.п.). Обзор систем ARIS Toolset, BPWin, ERWin и т.п. Стандартизация технических решений, шаблоны (образцы, patterns).</p>

4. Образовательные технологии

№ п/п	Наименование раздела	Виды учебных занятий	Образовательные технологии
1	2	3	4
1.	Развитие выразительных и инструментальных средств программирования.	<p>Лекции 1, 2 Лабораторное занятие 1</p> <p>Самостоятельная работа</p>	<p>Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач.</p> <p>Консультирование посредством электронной почты</p>
2.	Программа = алгоритм + данные.	<p>Лекция 3 Лабораторные занятия 2-16</p> <p>Самостоятельная</p>	<p>Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач.</p> <p>Консультирование посредством</p>

		я работа	электронной почты
3	Пакеты решений. Обзор средств проектирования программных приложений.	Лекции 5, 6 Самостоятельна я работа	Теоретическая лекция. Семинар-обсуждение. Консультирование посредством электронной почты

5. Оценка планируемых результатов обучения

5.1. Система оценивания

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль: • Опрос (1—2) • контроль за программным выполнением практических заданий (темы 3—5) • контр. работа (тема 3)	3—16 недели	5 баллов	5 баллов 45 баллов
	2—16 недели		
Промежуточная аттестация (зачет)	12 неделя	20 баллов	10 баллов
Итого за семестр (дисциплину)	17 неделя		40 баллов
Текущий контроль: • контроль за программным выполнением практических заданий (темы 6,7) • учебный проект (тема 8)			100 баллов
Промежуточная аттестация (экзамен)	7—16 недели	40 баллов	20 баллов
Итого за семестр (дисциплину)	17 неделя		40 баллов
			100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82			C
56 – 67			D
50 – 55	удовлетворительно	не зачтено	E
20 – 49			FX
0 – 19			F

5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ А,В	«отлично»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - грамотно выполнил большинство (в том числе и все обязательные) практические задания (программные образцы – samples); - глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации; - исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения; - свободно ориентируется в учебной и профессиональной литературе. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ С	«хорошо»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - грамотно, возможно с помощью преподавателя, выполнил достаточное количество практических заданий (программные образцы – samples), в том числе и все обязательные; - знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей; - правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами; - достаточно хорошо ориентируется в учебной и профессиональной литературе. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>

67-50/ D,E	«удовлетворительно»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - выполнил (возможно с помощью преподавателя) достаточное количество практических заданий (программные образцы – samples); - знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации; - испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами; - демонстрирует достаточный уровень знания учебной литературы по дисциплине. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не явился по неуважительной/неизвестной причине на аттестацию или:</p> <ul style="list-style-type: none"> - не выполнил (несмотря на возможную помощь преподавателя) достаточное количество практических заданий (программные образцы – samples); - не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. - испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами; - демонстрирует фрагментарные знания учебной литературы по дисциплине. <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

5.3.1. Образцы заданий для самостоятельного выполнения

Основным видом практической деятельности студентов является программное решение задач. Их можно разделить на следующие виды:

- обзоры выразительных и функциональных средств изучаемых языков и библиотек;
- программные образцы (samples) изучаемых элементов технологий и средств разработки ПО;
- традиционно «жестко», четко сформулированные задачи;
- совместное решение некоторых задач в рамках так называемых учебных проектов.

Студент выполняет обзор, следуя

- плану руководства пользователя (User Guide), которое, как правило, чаще всего является частью встроенной в IDE справочной подсистемы, или иного, аналогичного по жанру технического документа или литературного источника;
- указаниям преподавателя.

(Руководство пользователя прежде всего предназначено для освоения соответствующего средства программирования, в отличие от справочника по языку (language reference) или справочника (manual) иного программного средства.)

Обзор представляет собой программу, выводящую в виде протокола результаты применения команд к данным различных встроенных типов, использования различных управляющих конструкций и т.п.

Программные образцы отличаются от обычно рассматриваемых в курсах программирования четко сформулированных задач тем, что учащийся должен сам придумать «сюжет», сценарий выполнения программы, стремясь к тому, чтобы как можно проще, но, в то же время ясно, полно, «выпукло» показать эффект, результат демонстрируемого элемента технологии и/или средства программирования, ориентируясь на подобные образцы известных справочных систем.

Важную роль в формировании у студента навыков профессионального программиста играет такой вид деятельности, как проверка правильности выполнения достаточно сложных образцов своими коллегами. Конечно, в этом случае студент уже должен был решить соответствующую задачу и получить по ней у преподавателя зачет. Преподаватель же потом проверяет как правильность выполнения образца исполнителем, так и качество проверки проверяющим.

В рамках учебных проектов студенты или индивидуально, или объединенные на время в небольшие рабочие группы (аналоги малых коллективов разработчиков ПО) решают некоторые задачи программной реализации совместно с преподавателем. Здесь преподаватель, в зависимости от ситуации, может выступать в различных ролях: заказчика, менеджера проекта, реже – системного аналитика или системного архитектора. В роли менеджера проекта он контролирует выполнение отдельных частей проекта, следит за своевременностью выполнения последовательных его стадий (этапов). Студент же выступает здесь в качестве исполнителя – разработчика ПО и QA-инженера (от Quality Assurance - специалист по качеству ПО, «тестировщик»), а иногда – также в роли системного аналитика или системного архитектора.

Выполнение некоторых проектов разными коллективами обсуждается затем на занятии всей группой.

Почти все практические задания выполняются студентами в процессе самостоятельной работы.

Самостоятельная работа студента включает

- усвоение нового материала предыдущих занятий;
- подготовку к следующему занятию (в том числе самостоятельный предварительный разбор некоторой части его материала);
- подготовку докладов и кратких сообщений;

- выполнение домашних заданий (в основном это программные образцы изучаемых элементов технологий и средств разработки ПО);
 - выполнение некоторых небольших учебных проектов (см. п. 3) в составе рабочих групп;
 - подготовку к контрольным мероприятиям и промежуточной аттестации (экзамену). Все эти виды образовательной деятельности учащегося обеспечиваются
 - изучением источников из списка учебной литературы (см. п. 6), соответствующие разделы которых задаются преподавателем и усвоение которых контролируется преподавателем в ходе обсуждений и опросов на последующих занятиях;
 - использованием справочных подсистем, встроенных в применяемые программные средства (в частности, в IDE – интегрированные программные среды для разработки ПО);
 - использованием представленных в сети Интернет ресурсов, содержащих справочную информацию и техническую документацию
- Кроме того, студенты, по мере необходимости, получают указания преподавателя в виде планов выполнения практических заданий или фиксации в них ошибок, «недоделок», и т.п. Учащиеся также могут обращаться к преподавателю за получением консультаций. Такого рода контакты студента с преподавателем осуществляются как в аудитории, так и по электронной почте. При необходимости проводятся on-line конференции в Zoom.

Примеры зачетных практических заданий (программных образцов).

1. Для изучаемых в рамках данного курса языков программирования обзоры:
 - базовых «процедурных» средств языка;
 - представленных в языке средств ООП.
2. Выполняемые на C++ с использованием системной библиотеки поддерживаемых .NET Framework классов учебные проекты.
 - Управление выполнением внешнего процесса.

По сценарию `sample-a` запускается внешний исполняемый модуль (`exe`-файл), который, например, что-то пишет в известный «выходной» файл. «Обнаруживается» момент окончания выполнения этого процесса, после чего результат его работы считывается из его выходного файла и предъявляется пользователю образца.

 - Пример взаимодействия клиент-сервер.

В качестве сервера используется директория (папка), определенная как `localhost`. Это позволяет выполнять `sample` на единственном (локальном) компьютере.

На сервере размещается база данных и ее программная оболочка. Простая БД реализована в IDE MS Access. Это упрощение обусловлено невозможностью (из-за предоставленного для него объема времени) в рамках данного курса устанавливать на компьютере учащегося и осваивать MS SQL Server.

Простое приложение (`html`-страница) – поддерживаемое на стороне клиента браузером MS Internet Explorer или MS Edge запрашивает серверную БД и получает результат запроса через ASP.NET. Результат запроса предъявляется на странице пользователю клиента. На сервере запрос к БД выполняется через ADO.NET.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

Замечание: многие источники полностью или в виде актуальных фрагментов имеются в электронной библиотеке (ЭБ) Учебно-методического кабинета (УМК) Отделения интеллектуальных систем (ОИС), папка [d:_ois_lib] (либо [c:_ois_lib] в случае единственного логического диска) на компьютерах деканата.

а) Основная литература

1. Страуструп, Б. Дизайн и эволюция С++ [Электронный ресурс] / Б. Страуструп; Пер. с англ. - М.: ДМК Пресс, 2007. - 448 с.: ил. - (Серия «Для программистов»). - ISBN 5-94074-005-7.
<http://znanium.com/catalog.php?bookinfo=409529>
2. Мейерс С. Наиболее эффективное использование С++. 35 новых рекомендаций по улучшению ваших программ и проектов [Электронный ресурс]: учебное пособие. — Электрон. дан. — М.: ДМК Пресс, 2007. — 294 с.
<http://znanium.com/bookread2.php?book=407506>
2. Книга (документация) по Visual Studio 2015 (содержит ссылку на документацию по Visual Studio 2017) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/dd831853.aspx>
3. *Страуструп Б.* Язык программирования С++. Специальное издание. – М.: Издательство БИНОМ, 2011.– 1136 с.
(ЭБ УМК ОИС: [d:_ois_lib\2cpp\B_Stroustrup_Yazyk_programmirovaniya_C++ (2011).djvu])
4. Visual С++ в Visual Studio 2015. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx>
5. С#. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/kx37x362.aspx>
6. *Шилдт Г.* С# 4.0: полное руководство. – М.: ООО «И. Д. Вильямс», 2011 – 1056 с.
(ЭБ УМК ОИС: [d:_ois_lib\C-sh\GShildt-C#4.0.Polnoe_rukov-2011.pdf])
7. .NET Framework 4.6 и 4.5. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
[https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx)
8. Справочник по языку JavaScript. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
[https://msdn.microsoft.com/ru-ru/library/d1et7k7c\(v=vs.94\).aspx](https://msdn.microsoft.com/ru-ru/library/d1et7k7c(v=vs.94).aspx)
9. Начало работы с Python. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/dn705848.aspx>
10. *Рамбо Дж, Якобсон А., Буч Г.* UML: специальный справочник. – СПб: Питер, 2002.
(ЭБ УМК ОИС: [d:_ois_lib\progr-conc\UML\lang_UML._user-guid.pdf])
11. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд. – М.: «Издательство Бином», СПб: «Невский диалект» 1999.
(ЭБ УМК ОИС: [d:_ois_lib\progr-conc\UML\gradi_buch_- obektno-orientirovannuyu_analiz_i_proektirovanie_s_primerami_prilozheniy_na_s.pdf])
12. *Маклаков С.В.* BPWin и ERWin. Case-средства разработки информационных систем. М.: Диалог-МИФИ, 2000.
(ЭБ УМК ОИС: [d:_ois_lib\progr-conc\case\KnigBPwin_end_ERwin.zip])
13. *Константайн Л., Локвуд Л.,* Разработка программного обеспечения. – СПб: Питер, 2004.

(ЭБ УМК ОИС: [d:_ois_lib\progr-conc\razrabotka_progr_obesp.djvu])

14. Смит, Джейсон Мак-Колм. Элементарные шаблоны проектирования. — М. : ООО «И.Д. Вильямс», 2013.

(ЭБ УМК ОИС: [d:_ois_lib\progr-conc\patterns\Dzheyson_Mak-Kolm_Smit_Elementarnye_shablonyi_proektirovaniya.pdf])

б) Дополнительная литература

15. MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО: <http://msdn.microsoft.com>

16. Оберг Р. Дж. Технология COM+. Основы и программирование. – М.: Издательский дом «Вильямс», 2000. (ЭБ УМК ОИС: [d:_ois_lib\3wp\complus.djvu])

17. Прата С. Язык программирования C++. – М.: Издательский дом «Вильямс», 2007. (ЭБ УМК ОИС: [d:_ois_lib\2cpp\Prata_Yazyk-C++2007.pdf.djvu])

18. Коэн И. Полный справочник по HTML, CSS и JavaScript. Серия: Справочник профессионала. – М.: Эком Паблишера, 2007

19. Крейн Д., Пакарелло, Э., Джеймс, Д. Аjax в действии. – М.: Издательский дом «Вильямс», 2006.

20. Айзекс С. Dynamic HTML. – СПб.: BHV-Санкт-Петербург, 1998.

(ЭБ УМК ОИС: – с примерами в папке [d:_ois_lib\3ip\DHTML-book(byS_Isaacs)])

21. Бибо, Б., Кау, И. jQuery. Подробное руководство по продвинутому JavaScript – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 448 с.

(ЭБ УМК ОИС: [d:_ois_lib\3ip\jQuery_PodrobRukovodPoProdvinutomuJavaScript_2ed.pdf])

22. XML (Extensible Markup Language) – техническая документация Microsoft XML Parser SDK

23. XML Schema., – рабочий документ (рекомендация) консорциума W3C, опубликован (регулярно обновляется) на сайте консорциума (<http://www.w3.org/>): <http://www.w3.org/TR/xmlschema/>.

6.2 Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины

<http://msdn.microsoft.com> – MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО:

<https://msdn.microsoft.com/ru-ru/library/dd831853.aspx> – книга (документация) по Visual Studio 2015 (содержит ссылку на документацию по Visual Studio 2017) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx> – Visual C++ в Visual Studio 2015. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://msdn.microsoft.com/ru-ru/library/kx37x362.aspx> – C#. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

[https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx) – .NET Framework 4.6 и 4.5. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

[https://msdn.microsoft.com/ru-ru/library/d1et7k7c\(v=vs.94\).aspx](https://msdn.microsoft.com/ru-ru/library/d1et7k7c(v=vs.94).aspx) – Справочник по языку JavaScript. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://msdn.microsoft.com/ru-ru/library/dn705848.aspx> – Начало работы с Python. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://github.com/> – GitHub – веб-сервис для хостинга IT-проектов и их совместной разработки

<http://www.w3.org/TR/xmlschema/> – XML Schema (XSD) – рабочий документ (рекомендация) консорциума W3C

Перечень БД и ИСС

№п /п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2023 г. Web of Science Scopus
2	Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2023 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis
3	Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru
4	Компьютерные справочные правовые системы Консультант Плюс, Гарант

7. Материально-техническое обеспечение дисциплины

Занятия проводятся в компьютерных классах ауд. 307 и 706, расположенных по адресу 125993, Москва, Миусская пл., д. 6, стр.2,.

Этот компьютерный класс оснащен

- достаточным количеством объединенных в локальную сеть рабочих станций,
- медиапроектором и экраном,
- меловой доской.

В классе имеются возможности

- подключения ноутбука к медиапроектору,
- одновременного доступа в Интернет для преподавателя и студентов.

1. Перечень ПО

№п /п	Наименование ПО	Производитель	Способ распространения (лицензионное или свободно распространяемое)
1	Adobe Master Collection CS4	Adobe	лицензионное
2	ОС «Альт Образование» 8	ООО «Базальт СПО	лицензионное
3	Windows 10 Pro	Microsoft	лицензионное
4	Kaspersky Endpoint Security	Kaspersky	лицензионное
5	Microsoft Office 2016	Microsoft	лицензионное

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
 - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
 - письменные задания выполняются на компьютере в письменной форме;

- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата:
- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
 - в форме аудиофайла.
- для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа;
 - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
 - устройством для сканирования и чтения с камерой SARA CE;
 - дисплеем Брайля PAC Mate 20;
 - принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих:
 - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - передвижными, регулируемые эргономическими партами СИ-1;
 - компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

Лабораторные занятия по «блокам» схожих тем однотипны, они проводятся по единой схеме. На лекциях объясняются основные теоретические положения изучаемых тем,

формулируются постановки для части задач, предлагаются методы, возможные сценарии и технологические элементы для их решения. Часть задач объясняется непосредственно на практических занятиях.

Углубленно текущий материал изучается студентами самостоятельно.

Почти все практические задания также выполняются студентами в процессе самостоятельной работы.

На практических занятиях при необходимости проводится разбор текущего материала и контроль в форме опросов и дискуссий его усвоения.

Наконец, на практических занятиях осуществляется контроль за выполнением практических заданий, при необходимости проводятся индивидуальные или групповые консультации.

(Конкретнее эти положения изложены в п. 5.3.)

9.1. Примеры планов лабораторных занятий

1. Изучение средств программирования (темы 1, 2).

Язык C++: основные средства

Цель занятий: изучение базовых «процедурных» средств языка (т.е. исключая встроенную в язык подсистему ООП)

Форма проведения – опрос, дискуссии, решение задач.

Вопросы для обсуждения:

Синтаксис, виды команд, управляющие инструкции.

Типы и объявления.

Указатели и работа с ними.

Массивы и структуры.

Обзор операторов, выражения. Функции.

Перегруженные функции.

Пространства имен. Исключения.

Разработка исходного кода, компиляция, сборка программы.)

Примеры контрольных вопросов:

1. Парадигма процедурного программирования, в том числе на примере C++.
2. Подмножество C в C++ и расширения C к C++.
3. Применение рассматриваемых типов данных.
4. Элементы программирования динамических структур данных с использованием указателей и структур.

Задание:

программно реализовать обзорный образец базовых «процедурных» средств языка.

Список источников и литературы:

1. Страуструп, Б. Дизайн и эволюция C++ [Электронный ресурс] / Б. Страуструп; Пер. с англ. - М.: ДМК Пресс, 2007. - 448 с.: ил. - (Серия «Для программистов»). - ISBN 5-94074-005-7.

<http://znanium.com/catalog.php?bookinfo=409529>

2. Мейерс С. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов [Электронный ресурс]: учебное пособие. — Электрон. дан. — М.: ДМК Пресс, 2007. — 294 с.

<http://znanium.com/bookread2.php?book=407506>

3. *Страуструп Б.* Язык программирования C++. Специальное издание. – М.: Издательство БИНОМ, 2011.– 1136 с.

(ЭБ УМК ОИС: [d:_ois_lib\2cpp\B_Stroustrup_Yazyk_programirovaniya_C++ (2011).djvu])

4. Visual C++ в Visual Studio 2015. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):

<https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx>

2. Выполняемые на C++ с использованием системной библиотеки поддерживаемых .NET Framework классов учебные проекты. (тема 2).

Разработка распределенных приложений на платформе .NET

Цель занятий: изучение архитектуры «клиент-сервер» и пример реализации с использованием соответствующего подмножества классов из системной библиотеки.NET.

Форма проведения – опрос, дискуссии, решение задач.

Вопросы для обсуждения:

Классы ADO.NET.

Классы ASP.NET.

HTTP-серверы, служба IIS и организация localhost.

Общая архитектура образца и сценарий его выполнения.

Пример (зачетного) задания:

Пример взаимодействия клиент-сервер.

В качестве сервера используется директория (папка), определенная как localhost. Это позволяет выполнять sample на единственном (локальном) компьютере.

На сервере размещается база данных и ее программная оболочка. Простая БД реализована в IDE MS Access. Это упрощение обусловлено невозможностью (из-за предоставленного для него объема времени) в рамках данного курса устанавливать на компьютере учащегося и осваивать MS SQL Server.

Простое приложение (html-страница) – поддерживаемое на стороне клиента браузером MS Internet Explorer или MS Edge запрашивает серверную БД и получает результат запроса через ASP.NET. Результат запроса предьявляется на странице пользователю клиента. На сервере запрос к БД выполняется через ADO.NET.

Список источников и литературы:

1. \.NET Framework 4.6 и 4.5. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):

[https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx)

2. MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО: <http://msdn.microsoft.com>

АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина «Концепции и технологии современного программирования» реализуется на Отделении интеллектуальных систем в гуманитарной сфере Учебнонаучным центром программного и лингвистического обеспечения интеллектуальных систем в 8 семестре.

Цели дисциплины.

Основных целей курса две:

- дать студенту целостное представление о концепциях и технологиях современного программирования, тем самым, резюмируя и обобщая его знания, полученные при изучении всех других дисциплин по направлению «Программирование».
- способствовать дальнейшему формированию у студента навыков работы самостоятельного программиста.

Задачи дисциплины.

Для этого решаются следующие образовательные задачи:

в рамках данного курса предлагается обзор основных концепций и технологий, применяемых в языках программирования и других средствах программирования, в частности, реализуемых в современных инструментальных системах проектирования и программирования, рассматриваются тенденции дальнейшего развития выразительных средств и технологий.

Дисциплина направлена на формирование следующих компетенций:

ПК-1 – Способен разрабатывать алгоритмы обработки информации с использованием современных математических методов

ПК-3 – Способен разрабатывать и тестировать новые программы и интерфейсы систем.

В результате освоения дисциплины (*модуля*) обучающийся должен:

Знать.

- Студенты должны овладеть следующими общими понятиями, связанными с технологиями современного программирования и развитием поддерживающих эти технологии программных средств:
 - предметная область, прикладная задача, приложение;
 - модель (предметной области, приложения);
 - «метафора» (совокупность идей, положенных в основу) технологии/средства;
 - основанный на воплощении метафоры «подход к решению» (задачи), стиль программирования (задачи);
 - реализующие некоторую метафору (см. выше) адекватные средства программирования (возможно, выразительные средства в языке программирования).
- Студенты должны иметь представление:
 - об истории, современном состоянии и тенденциях развития основных концепций, технологий и средств (включая языки) программирования;
 - об архитектуре программных (в том числе распределенных) приложений;
 - представление (на элементарном, популярном, начальном уровне) об инструментальных средствах проектирования программных приложений.
- Знать основные принципы использования интегрированных сред разработки (IDE) программных приложений.

Уметь:

- грамотно разрабатывать алгоритмы и их программные решения, комплексно используя применяемые для разработки адекватные инструментальные средства;
- по возможности, адекватно, выбирать средства для разработки программных приложений, исходя из особенностей прикладной задачи и предметной области.

Владеть навыками:

- анализа документации к программным системам;
- программиста, умеющего разбираться в новых для себя средствах программирования;
- самостоятельного специалиста в области программной реализации прикладных приложений.

Общая трудоемкость освоения дисциплины составляет 3 зачетные единицы.

