

МИНОБРНАУКИ РОССИИ



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

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ГУМАНИТАРНОЙ СФЕРЕ
Кафедра математики, логики и интеллектуальных систем в гуманитарной сфере

МЕТОДЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО 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

Методы объектно-ориентированного 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 Цель 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. Цель и задачи дисциплины

Цель дисциплины:

научить студентов основам работы со сложными структурами данных с использованием средств объектно-ориентированного программирования (ООП).

Задачи дисциплины:

- овладение студентами системой алгоритмических понятий и навыков, необходимых для формирования общей культуры будущего специалиста по информационным системам и дальнейшего углублённого изучения теоретических основ и практических методов построения систем искусственного интеллекта;
- изучение возможностей использования структур данных в программировании;
- ознакомление студентов с применением основных принципов ООП на некоторых относительно несложных, но нетривиальных примерах структур данных;
- подготовка студентов к дальнейшему углублённому изучению возможностей применения объектно-ориентированного подхода;
- развитие навыков применения изученного математического аппарата к решению практических задач.

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

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
<p>ПК-1 Способен разрабатывать алгоритмы обработки информации с использованием современных математических методов</p> <p>ПК-9 Способен применять методы анализа вариантов, разработки и поиска компромиссных решений</p>	<p>ПК-1.1 Знает теоретические основы построения алгоритмов обработки информации</p> <p>ПК-1.2 Умеет описывать алгоритмы обработки информации с использованием современных математических методов</p> <p>ПК-1.3 Имеет практический опыт разработки алгоритмов обработки информации с использованием современных математических методов</p> <p>ПК-9.1 Знает теоретические основы методов оптимизации</p> <p>ПК-9.2 Умеет применять методы поиска оптимальных решений в практической деятельности</p> <p>ПК-9.3 Имеет практический опыт участия в анализе преимуществ и рисков возможных решений с использованием математических методов</p>	<p><i>Знать:</i></p> <ul style="list-style-type: none"> ● способы представления сложных структур данных в разных языках программирования (списки, деревья, графы и т.д.). ● круг задач, рассматриваемых в объектно-ориентированном программировании; ● понятия объекта и класса; ● методы и их наследование; ● понятие полиморфизма; ● как представляются эти понятия в разных языках программирования. <p><i>Уметь:</i></p> <ul style="list-style-type: none"> ● сравнивать разные языки программирования с точки зрения использования ими сложных структур данных и средств ООП; ● писать программы обработки нетривиальных структур данных в рамках ООП. <p><i>Владеть</i> применением основных средств ООП (классов, методов и т.д.) при разработке несложных приложений искусственного интеллекта.</p>
<p>ПК-8 Способен разрабатывать техническую документацию и использовать средства автоматизации при проектировании</p>	<p>ПК-8.1 Знает стандарты на техническую документацию</p> <p>ПК-8.2 Умеет применять CASE-технологии при проектировании</p>	<p><i>Знать</i> основные принципы использования интегрированных сред разработки (IDE) программных приложений.</p> <p><i>Уметь</i> грамотно разрабатывать алгоритмы и их программные</p>

информационных систем и систем, основанных на знаниях	информационных систем и систем, основанных на знаниях и отображать результаты проектирования в технической документации ПК-8.3 Имеет практический опыт участия в разработке технической документации и проектировании информационных систем и систем, основанных на знаниях;	решения, комплексно используя применяемые для разработки адекватные инструментальные средства. <i>Владеть</i> навыками: анализа документации к программным системам; работы в средах разработки приложений ООП (в основном, на примере Microsoft Visual Studio C++).
---	---	---

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Методы объектно-ориентированного программирования» относится к формируемой участниками образовательных отношений части блока Б1 дисциплин учебного плана и включена в блок Б1.В.ДВ.06 элективных дисциплин.

Предполагается, что приступающий к изучению настоящей дисциплины студент успешно освоил курсы для бакалавров: «Алгебра», «Программирование на СИ».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: все другие курсы для бакалавров по направлению «Программирование», а также «сопутствующие» им, изучаемые начиная с 3-го семестра; выполнение ВКР бакалавра, а также связанные с программированием дисциплины, изучаемые в магистратуре ОИС в ГС.

2. Структура дисциплины

Структура дисциплины (модуля) для очной формы обучения

Общая трудоёмкость дисциплины составляет 2 з.е., 72 ч., в том числе контактная работа обучающихся с преподавателем 28 ч., самостоятельная работа обучающихся 44 ч.

№ п/п	Раздел дисциплины/темы	Се м е ст р	Виды учебной работы (в часах)					Формы текущего контроля успеваемости, форма промежуточной аттестации (<i>по семестрам</i>)	
			Контактная				Про меж уточ ная атте стац ия		Са мо сто яте льная раб ота
			Лек ции	Се ми нар	Прак тичес кие занят ия	Лабора тор ные занят ия			
1.	История	3	2				4	Опрос	

	появления ООП в контексте общего развития технологий и средств программирования								
2.	Основные понятия и принципы ООП	3	2					4	Опрос, оценка выполнения практических заданий
3	Воплощение принципов ООП в языке C++. Инструментальная среда разработки приложений (IDE) Microsoft Visual Studio C++	3			2			2	Опрос, оценка выполнения практических заданий
4	Структуры данных в программировании и возможные способы их реализации.	3	2		2			6	Опрос, оценка выполнения практических заданий, контрольная работа
5	Представления множеств. Сортировки и поиск.	3	4					4	Опрос, оценка выполнения практических заданий, контрольная работа
6	Учебный проект	3	2		12			24	Опрос, оценка выполнения практических заданий,
	Итоговая аттестация	3							зачет
	Итого		12		16			44	

3. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	История появления ООП в контексте общего развития технологий и средств программирования	Языки Симула, Смолток, Паскаль, Более современные языки: C++, Java, C#, Выразительные средства декомпозиции задач в программных системах. Обзор основных стилей программирования (процедурное, функциональное, логическое)

		<p>программирование).</p> <p>Обзор парадигм и технологий программирования: историческая ретроспектива, настоящее время, тенденции развития.</p> <p>От модульного программирования к ООП. Парадигмы ООП. Виды ООП.</p>
2.	<p>Основные понятия и принципы ООП</p>	<p>Понятия объекта и класса.</p> <p>Инкапсуляция, наследование, полиморфизм.</p> <p>Определения классов. Объекты. Множественное наследование. Методы и обобщенные функции.</p> <p>Примеры реализации классов структур данных в ООП.</p> <p>Системные функции для работы с объектами и определениями классов.</p> <p>- визуальное программирование (обзор IDE, декларативное определение элементов интерфейса разрабатываемой программы и т.п.)</p> <p>...</p>
3	<p>Воплощение принципов ООП в языке C++.</p> <p>Инструментальная среда разработки приложений (IDE) Microsoft Visual Studio C++</p>	<p>Объекты, классы, наследование в языке C++.</p> <p>Введение в систему MS Visual Studio C++: установка и вводные упражнения на разработку программ в ней.</p>
4	<p>Структуры данных в программировании и возможные способы их реализации.</p>	<p>Типы данных: простые и составные, встроенные и определяемые программистом, статические и динамические, проч. классификации.</p> <p>Принцип локальности в программировании и реализация оболочек данных, основная функциональность последних.</p> <p>Основные динамические структуры данных (списки, динамические массивы, множества, стеки, очереди, деки, графы (в частности, деревья и сети) и возможные способы их реализации (процедурные языки).</p> <p>Объектные модели. Объектный подход к реализации структур данных.</p>
5	<p>Представления множеств. Сортировки и поиск.</p>	<p>Понятие о вычислительной сложности алгоритмов.</p> <p>Множества неупорядоченные, упорядоченные, представленные функцией принадлежности как подмножества фиксированного универсума. Операции над множествами в этих представлениях.</p> <p>Задача сортировки. Сортировки специальные и универсальные (сравнениями). Сложность.</p> <p>Линейная лексикографическая сортировка.</p> <p>Сортировки сравнениями; «простые» квадратичные, слиянием, деревом, быстрая.</p> <p>Двоичный поиск. Двоичное дерево. Его изменение и балансировка.</p>
6	<p>Учебный проект</p>	<p>Объектная реализация «библиотеки», включающей стек, очередь, деку, в виде иерархии (по наследованию)</p>

4. Образовательные технологии

№ п/п	Наименование раздела	Виды учебных занятий	Образовательные технологии
1	2	3	4
1.	История появления ООП в контексте общего развития технологий и средств программирования	Лекция 1 Самостоятельная работа	Теоретическая лекция. Семинар-обсуждение. Консультирование посредством электронной почты
2.	Основные понятия и принципы ООП	Лекция 2 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
3	Воплощение принципов ООП в языке C++. Инструментальная среда разработки приложений (IDE) Microsoft Visual Studio C++	Практическое занятие 1 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
4	Структуры данных в программировании и возможные способы их реализации.	Лекция 3 Практическое занятие 2 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
5	Представления множеств. Сортировки и поиск.	Лекции 4,5 Самостоятельная работа	Теоретическая лекция. Семинар-обсуждение. Консультирование посредством электронной почты
6	Учебный проект	Лекция 6 Практические занятия 3-8 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты

5. Оценка планируемых результатов обучения

5.1. Система оценивания

<i>Форма контроля</i>	<i>Срок отчетности</i>	<i>Макс. количество баллов</i>	
		<i>За одну работу</i>	<i>Всего</i>
Текущий контроль: • Опрос (1—2) • контроль за программным выполнением практических заданий (темы 3—5) • контр. работа (тема 3)	3—16 недели 2—16 недели 12 неделя	5 баллов 20 баллов	5 баллов 45 баллов 10 баллов
Промежуточная аттестация (зачет)	17 неделя		40 баллов
Итого за семестр (дисциплину)			100 баллов
Текущий контроль: • контроль за программным выполнением практических заданий (темы 6,7) • учебный проект (тема 8)	7—16 недели	40 баллов	20 баллов 40 баллов
Промежуточная аттестация (экзамен)	17 неделя		40 баллов
Итого за семестр (дисциплину)			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	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	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/ A,B	«отлично»/ «зачтено»	Выставляется обучающемуся, если он: - грамотно выполнил большинство (в том числе и все

		<p>обязательные) практические задания (программные образцы – samples);</p> <ul style="list-style-type: none"> - 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

		<p>задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами;</p> <p>- демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <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> <p>- не выполнил (несмотря на возможную помощь преподавателя) достаточное количество практических заданий (программные образцы – samples);</p> <p>- не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>- испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами;</p> <p>- демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <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, режиссера – 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);
- 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 C++;
- представленных в языке C++ средств ООП.

Образцы заданий для самостоятельного выполнения

1. Написать программу отправления приветствия списку друзей
2. Реализовать тип коллекций с помощью массивов и указателей.
3. Реализовать сбалансированные деревья с помощью связанных списков.

Образцы заданий для контрольной работы

Примеры контрольных работ:

Контрольная работа 1

Реализовать в языке C++ средствами ООП программу представления и пополнения линейного списка.

Контрольная работа 2

Рекурсивно реализовать в языке C++ сортировку слиянием массива.

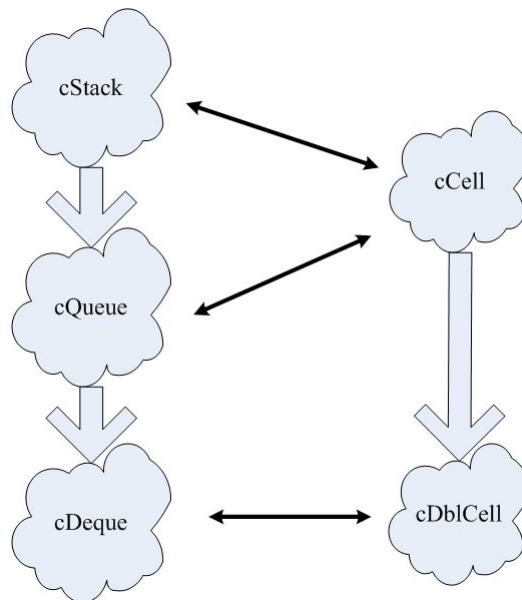
Спецификация учебного проекта (тема 6).

Объектно реализуются следующие структуры данных:

- ячейки для хранения данных
 - в стеке и очереди,
 - в деке
- оболочки данных для
 - стека,
 - очереди,
 - деки.

Важно «методически» - доступ к данным во всем примере только через аксессоры!

Возникают две иерархии по наследованию взаимосвязанных классов:



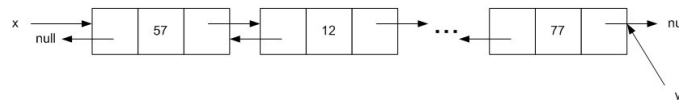
Ячейки.

Методы ячеек - это только ф-ии доступа к данным.

В стеке и очереди данные сохраняются в односвязном линейном списке:

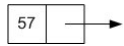


В деке эту роль выполняет двусвязный список, доступный для изменения с обоих концов:



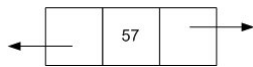
Для спецификации классов приводим примеры «хэдэров» (содержимое .h файлов).

Класс cCell



```
class cCell {  
  
protected:  
int data;  
cCell* next;  
  
public:  
cCell(int d); //конструктор для ячейки  
  
void setData(int d); //задать данное для ячейки  
int getData(); //получить данное ячейки  
void setNext(cCell*p); //установить указатель на следующую ячейку  
virtual cCell* getNext(); //получить указатель на следующую ячейку  
  
void cPrintCell(); //печать списка  
  
};
```

Класс cDblCell

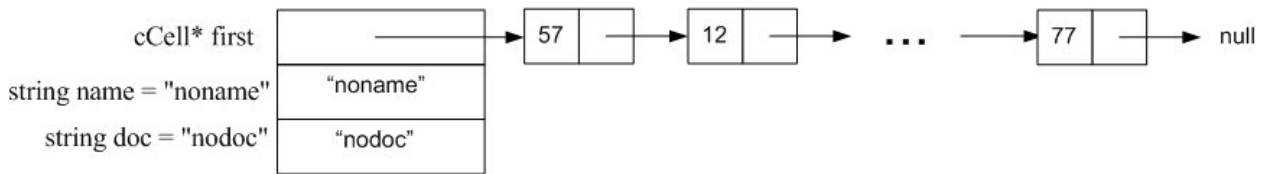


class cDblCell - подкласс cCell

```
protected:  
cDblCell* prev;  
  
public:  
cDblCell(int d); // конструктор для ячейки, наследуется от cCell(int d)  
  
void setPrev(cDblCell* p); //установить указатель на предыдущую ячейку  
cDblCell* getPrev(); //получить указатель на предыдущую ячейку  
  
cDblCell* getNext(); //получить новый указатель на следующую ячейку (типа cDblCell)
```

Обочки кроме списка эл-тов содержат (для примера) еще данные – имя и что-то вроде документирования (комментарий в памяти).

Класс cStack



```
class cStack
{
protected:
cCell* first;
string name = "noname";
string doc = "nodoc";

public:
cStack(); //конструктор для стека

//Аксесоры (методы доступа к данным)

void setFirst(cCell*c); //установить указатель на первый элемент списка
virtual cCell * getFirst(); //получить указатель на первый элемент списка
void setDoc(string dc); //установить поле doc
string getDoc(); //получить поле doc
void setName(string n); // установить имя
string getName(); // получить имя

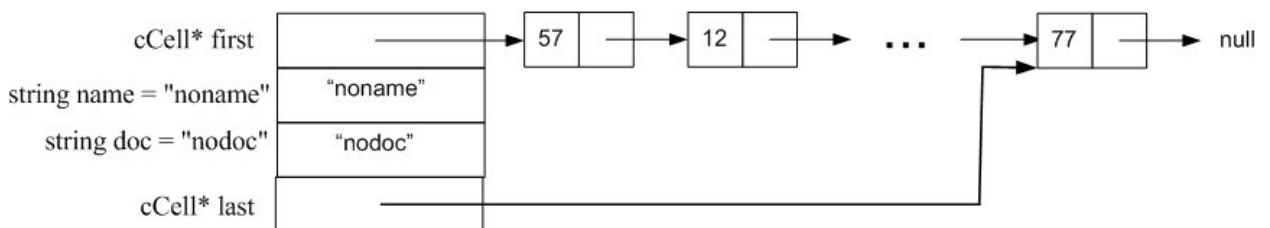
virtual bool isEmpty(); //проверка, пуст ли список

virtual void Push (int d); //добавление элемента в начало стека
virtual int Pop (); //удаление элемента из начала стека

int getF(); //получить data первого элемента не удаляя

};
```

Класс cQueue



class cQueue (дочерний) подкласс класса cStack

protected:

cCell* last;

public:

cQueue(); // конструктор для очереди, наследуется от cStack()

virtual cCell* getLast(); //получить указатель на последний элемент списка

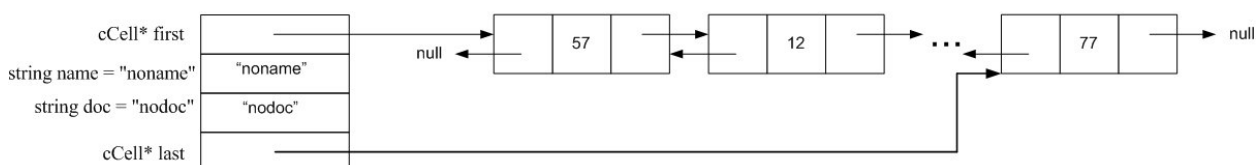
void setLast(cCell*last); //установить указатель на последний элемент списка

bool isEmpty(); //проверка, пуст ли список очереди

 // int Pop(); // удаление элемента из начала очереди

virtual void Push(int d); //добавление элемента в конец

Класс cDeque



Специфицируется студентами самостоятельно.

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. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979. К сожалению, эта книга больше не переиздавалась. Вместо нее можно использовать новый труд ее авторов, в основу которого положены ее первые шесть глав:

Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2000 (и последующие переиздания).

(издания 2001 и 2010 гг. имеются в библиотеке РГГУ)

По мнению автора курса, общие принципы и идеи, лежащие в основе так называемых базовых алгоритмов и обычно применяемые для решения алгоритмических задач в различных областях, а также изложение самих, рассматриваемых в данном курсе, базовых алгоритмов, предпочтительно изучать, используя в качестве учебного пособия книгу «Построение и анализ ...» этих авторов, вышедшую в русском переводе в 1979 году (имеется в электронной библиотеке Учебно-методического кабинета Отделения интеллектуальных систем (ОИС): (ЭБ УМК ОИС: [d:_ois_lib\1prg\alg\AhoHopcroftUllman_PostrIAnalizVychislitAlg-v.djvu])

6. Вирт Н. Алгоритмы и структуры данных, – СПб.: «Невский диалект», 2008.

Вышедшие в этом издательстве предыдущие переиздания (например, составителю известны переиздания 2005

(имеется в электронной библиотеке УМК ОИС: файлы

virt_n_algoritmy_i_strukturny_dannyh(ver2).djvu либо virt_niklaus_algoritmy_i_strukturny_dannyh.pdf в папке 1prg\alg\virt)

и 2001 годов (имеется в библиотеке РГГУ)) попрежнему актуальны, наряду с

Вирт Н. Алгоритмы и структуры данных, – М.: Мир, 1989,

(имеется в электронной библиотеке УМК ОИС:

(ЭБ УМК ОИС: [d:_ois_lib\1prg\alg\virt\Virt_n_algoritmy_i_strukturny_dannyh.djvu])

7. Липский В. Комбинаторика для программистов. – М.: Мир, 1988

(имеется в электронной библиотеке УМК ОИС: файл Lipsky_Kombinatorika_dlja_programmistov.pdf в папке 1prg\alg

(ЭБ УМК ОИС: [d:_ois_lib\1prg\alg\Lipsky_Kombinatorika_dlja_programmistov.pdf])

б) Дополнительная литература

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

9. Подбельский В. В. Практикум по программированию на языке C, – М.: «Финансы и статистика», 2004.

10. Подбельский В. В. Язык C++, – М.: «Финансы и статистика», 2007.

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

12. Лафоре Р. Объектноориентированное программирование в C++. Классика Computer Science, 4-е изд. – СПб.: «Невский диалект», 2008.

13. Б.Мейер, Объектно-ориентированное конструирование программных систем. М. :Издательский дом "Русская редакция", 2005.

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/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

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

Тема 2 (2 ч.) Основные понятия и принципы ООП

Цель занятия: Общее ознакомление с проблематикой ООП и структур данных

Форма проведения – обсуждение, опрос.

Вопросы для обсуждения:

Предпосылки появления ООП

Список источников и литературы:

- *Страуструп Б.* Язык программирования C++. Специальное издание. – М.: Издательство БИНОМ, 2011.
- *Б.Мейер*, Объектно-ориентированное конструирование программных систем. М.: Издательский дом "Русская редакция", 2005.

Материально-техническое обеспечение занятия: доска, компьютер + медиапроектор.

Тема 3 (2 ч.) Воплощение принципов ООП в языке C++. Инструментальная среда разработки приложений (IDE) Microsoft Visual Studio C++

Цель занятия: ознакомление с тем, как принципы ООП воплощаются в языке Java и среде программирования MS Visual Studio C++.

Форма проведения – обсуждение, опрос, практические упражнения.

Вопросы для обсуждения:

Способы определения классов. в MS Visual Studio C++

- *Список источников и литературы:*
- 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>

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

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>

Материально-техническое обеспечение занятия: компьютерный класс с доской и , медиапроектором.

АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина «Методы объектно-ориентированного программирования» реализуется на Отделении интеллектуальных систем в гуманитарной сфере кафедрой математики, логики и интеллектуальных систем в гуманитарной сфере в 3 семестре.

Цель дисциплины:

научить студентов основам работы со сложными структурами данных с использованием средств объектно-ориентированного программирования (ООП).

Задачи дисциплины:

- овладение студентами системой алгоритмических понятий и навыков, необходимых для формирования общей культуры будущего специалиста по информационным системам и дальнейшего углублённого изучения теоретических основ и практических методов построения систем искусственного интеллекта;
- изучение возможностей использования структур данных в программировании;
- ознакомление студентов с применением основных принципов ООП на некоторых относительно несложных, но нетривиальных примерах структур данных;
- подготовка студентов к дальнейшему углублённому изучению возможностей применения объектно-ориентированного подхода;
- развитие навыков применения изученного математического аппарата к решению практических задач.

Дисциплина направлена на формирование следующих компетенций:

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

ПК-8 – Способен разрабатывать техническую документацию и использовать средства автоматизации при проектировании информационных систем и систем, основанных на знаниях.

ПК-9 – Способен применять методы анализа вариантов, разработки и поиска компромиссных решений

В результате освоения дисциплины (*модуля*) обучающийся должен

Знать:

- способы представления сложных структур данных в разных языках программирования (списки, деревья, графы и т.д.).
- круг задач, рассматриваемых в объектно-ориентированном программировании;
- понятия объекта и класса;
- методы и их наследование;
- понятие полиморфизма;
- как представляются эти понятия в разных языках программирования.

Знать основные принципы использования интегрированных сред разработки (IDE) программных приложений.

Уметь:

- сравнивать разные языки программирования с точки зрения использования ими сложных структур данных и средств ООП;
- писать программы обработки нетривиальных структур данных в рамках ООП;
- грамотно разрабатывать алгоритмы и их программные решения, комплексно используя применяемые для разработки адекватные инструментальные средства.

Владеть навыками:

- применения основных средств ООП (классов, методов и т.д.) при разработке несложных приложений искусственного интеллекта;

- анализа документации к программным системам;
- работы в средах разработки приложений ООП (в основном, на примере Microsoft Visual Studio C++).

Общая трудоемкость освоения дисциплины составляет 2 зачетные единицы.

