

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент программной инженерии

**Рабочая программа дисциплины
Алгоритмы и структуры данных (часть 2)**

для образовательной программы «Программная инженерия»
направления подготовки 09.03.04 «Программная инженерия»
уровень - бакалавр

Разработчик программы

Дегтярев К.Ю., к.т.н., доцент, kdegtiarev@hse.ru

Одобрена на заседании департамента программной инженерии « ___ » _____ 2015 г.

Руководитель департамента Авдошин С.М. _____

Рекомендована Академическим советом образовательной программы

« ___ » _____ 2015 г., № протокола _____

Утверждена « ___ » _____ 2015 г.

Академический руководитель образовательной программы

Шилов В.В. _____

Москва, 2015

*Настоящая программа не может быть использована другими подразделениями университета
и другими вузами без разрешения подразделения-разработчика программы.*



1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины «Алгоритмы и структуры данных» (часть 2, модули 3-4) устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину (лекции и практические занятия), учебных ассистентов и студентов направления 09.03.04 «Программная инженерия» подготовки бакалавра, изучающих дисциплину «Алгоритмы и структуры данных» во втором семестре (модули 3 и 4) 2015-2016 учебного года.

Программа дисциплины разработана в соответствии с:

- образовательным стандартом (ОС) Федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики» по направлению 09.03.04 «Программная инженерия» подготовки бакалавра (действует с 2014 г.), разработанным в соответствии с Федеральным законом №273-ФЗ «Об образовании в Российской Федерации»; <http://www.hse.ru/standards/standard>,
- образовательной программой (базовым учебным планом, составленным на основе ОС ПИ) направления 09.03.04 «Программная инженерия» подготовки бакалавра;
- рабочим учебным планом (для 2-го года обучения; 2015-2016 учебный год) по направлению 09.03.04 «Программная инженерия» подготовки бакалавра, утвержденным в 2015 г. (<http://asav.hse.ru/plans.html?login=web&password=web>).

2 Цели освоения дисциплины

Цели освоения дисциплины «Алгоритмы и структуры данных» (часть 2, модули 3-4):

- формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области структур данных и теории алгоритмов, пониманием концепции абстрактных типов данных (АТД) и подходов к их реализации на языке C++ на основе принципов объектно-ориентированного построения программ, оценки влияния выбора структур данных и/или алгоритмов на производительность (быстродействие/эффективность) программ;
- получение практических навыков решения задач с использованием разных структур данных (напр., линейных списков, стеков, очередей, хэш-таблиц и пр.), используя концепции абстракции данных и модульного программирования;
- развитие умений, основанных на полученных теоретических знаниях, позволяющих на творческом и репродуктивном уровне предлагать и применять эффективные подходы к решению (алгоритмизации) поставленных задач с использованием данных простой и сложной структуры;
- получение студентами навыков *самостоятельной (исследовательской) работы*, предполагающей изучение специфических особенностей работы со структурами данных в рамках разработки подходов (алгоритмов) к решению поставленной задачи, вопросов управления памятью в C++ и использования компонентов стандартной библиотеки шаблонов (STL).

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:



- знать базовые абстрактные типы (структуры) данных (*контейнеры*), понимать их особенности, применимые операции и методы реализации АТД на языке С++; контейнеры, итераторы и алгоритмы как основные компоненты STL.
- уметь разрабатывать С++ программы, реализующие заданный алгоритм и использующий определенные структуры данных.
- иметь навыки (приобрести опыт) и владеть основами процедурного и объектно-ориентированного программирования на языке С++, работы с шаблонами функций и классов, использования STL как реализованную на языке С++ коллекцию обобщенных (*generic*) структур данных (*контейнеров*) и алгоритмов, методами оценки трудоемкости (сложности) алгоритмов, подходами к измерению времени в программных реализациях алгоритмов решения поставленных задач.

В результате изучения дисциплины студент осваивает (и развивает) следующие профессиональные компетенции:

1. Научно-исследовательская деятельность (*общие*):

- участие в проведении исследований (экспериментов, наблюдений и количественных измерений), связанных с объектами профессиональной деятельности (программными продуктами, проектами, ..., методами и инструментами программной инженерии) в соответствии с утвержденными заданиями и методиками

Формы и методы обучения, способствующие формированию и развитию компетенции: лекции и практические занятия, консультации и самостоятельная работа студентов (изучение теоретического материала и практическая реализация программ, предлагаемых в качестве основных и дополнительных заданий, подготовка отчетов);

2. Проектная деятельность (*отдельные*):

- участие в проектировании компонентов программного продукта в объеме, достаточном для их конструирования в рамках поставленного задания;
- создание компонент программного обеспечения (кодирование, отладка, модульное и интеграционное тестирование);
- выполнение измерений и рефакторинг кода в соответствии с планом

Формы и методы обучения, способствующие формированию и развитию компетенции: лекции и практические занятия, консультации и самостоятельная работа студентов (изучение теоретического материала и практическая реализация программ); выполнение домашних заданий; подготовка отчета о выполненной работе (анализ поставленной задачи, предлагаемые варианты решения, особенности реализации и пр.) и программ на языке С++ (кодирование, отладка и тестирование, рефакторинг);

3. Технологическая деятельность:

- освоение и применение средств проектирования, разработки и тестирования программного обеспечения;
- использование типовых методов для контроля, оценки и обеспечения качества программной продукции

4. Производственная деятельность (*отдельные*):

- участие в процессах разработки программного обеспечения

Формы и методы обучения, способствующие формированию и развитию компетенции: лекции и практические занятия, консультации и самостоятельная работа студентов (изучение теоретического материала и практическая реализация программ); выполнение домашних заданий; подготовка отчета о выполненной работе (анализ поставленной задачи, предлагаемые варианты решения, особенности реализации и пр.) и программ на языке С++ (кодирование, отладка и тестирование, рефакторинг).



4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к блоку профессионального цикла дисциплин, базовой части (**Б.ПЦ.Б**) рабочего учебного плана направления 09.03.04 «Программная инженерия» подготовки бакалавра (ОП «Программная инженерия», 2015–2016 учебный год, факультет компьютерных наук) – см. <http://asav.hse.ru/plans.html?login=web&password=web>.

В соответствии с рабочим учебным планом на 2015–2016 учебный год, дисциплина «Алгоритмы и структуры данных» (часть 2) предлагается студентам 2-го курса бакалавриата в 3-м и 4-м модулях учебного года. Количество аудиторных часов – **84** (44+40; на 10 и 11 учебных недель модулей 3 и 4, соответственно); эти часы делятся поровну (с учетом праздничных дней, установленных в РФ). Общее количество часов, отведенных студентам для самостоятельной работы – **80** (половина от того количества часов, которые предусмотрены РУП для двух частей 4-х модульной дисциплины «Алгоритмы и структуры данных» в 2015–2016 учебном году). Текущий контроль (второе полугодие, модули 3 и 4) – **(1)** одно домашнее задание (ДЗ) и **(1)** одна контрольная работа (КР). Итоговый контроль – **экзамен** в конце 4-го модуля. Общее число часов (часть 2) – **164**.

Изучение данной дисциплины базируется на знаниях, полученных студентами при освоении учебных дисциплин «Дискретная математика», «Программирование», «Алгоритмы и структуры данных» (часть 1) первого и второго годов обучения, основ программирования в части базовых алгоритмических конструкций. Таким образом, «Алгоритмы и структуры данных» (часть 2) можно рассматривать в качестве одной из важных составляющих цепочки предлагаемых в рамках БУП направления 09.03.04 дисциплин, связанных с теоретической информатикой, основами информационных технологий и программирования (первых двух лет обучения).

Дисциплина является основой для последующего изучения таких дисциплин как «Проектирование и архитектура программных систем», «Обеспечение качества и тестирование» (3-й курс), отдельных научно-исследовательских семинаров (3-й и 4-й курсы | цикл **Б.6** 'Научный семинар'), а также выполнения курсовых, проектных и выпускных работ, предусмотренных БУП по направлению подготовки 09.03.04.

●●● **Примечание:** Как мы знаем, алгоритмы решают возникающие повсеместно вычислительные задачи; естественно, мы всегда требуем от алгоритма получения правильного решения поставленной задачи. Разработка алгоритма является весьма трудоемкой задачей, требующей одновременного мониторинга **соответствия** предлагаемых **действий** сформулированным **целям** (верхний уровень представления проблемы) и отдельных мелких шагов (нижний уровень представления проблемы), образующих в совокупности этапы действий, которые и приводят в конечном итоге к решению задачи. Для описания последовательности шагов алгоритма обычно используются три формы его описания, а именно:

- **вербальная** (словесная) **форма**, которая может грешить неточностью и/или нечеткостью формулировок, серьезно влияющих на понимание представляемых действий,
- **псевдокод** (форма представления с «облегченным» синтаксисом, использующий смесь конструкций обычного (повседневного) языка и языка программирования) – как образно отмечено Стивенем Скиеной (**Steven Sol Skiena, Teaching Professor of Computer Science at Stony Brook University**), псевдокод – это «язык программирования, который никогда не выдает сообщений о синтаксических ошибках»,
- **язык программирования** (описание алгоритма по жестким синтаксическим правилам выбранного языка программирования, позволяющим достичь точного описания реализуемых шагов).

Последняя из перечисленных форм описания шагов **АЛГОРИТМА** представляется наиболее интересной и практически значимой, поскольку современная жизнь связана с повседневным использованием компьютеров (вычислительных устройств), на которых работают многочис-



ленные и разнообразные реализующие алгоритмы программы. Компьютеры призваны существенно ускорить выполняемые вычисления, хранить и обрабатывать большие объемы информации (**ДАННЫХ**), которые представляют собой **абстракцию** связанной с рассматриваемой проблемой реальности. Такие абстракции (по сути, **модели предметной области**) дают возможность описать данные, которыми оперируют алгоритмы.

Для обработки данных нам следует определить типы (данных) и множество операций, которые выполняются над этими данными, при этом детали того, как реализованы эти операции либо неизвестны, либо скрываются. Таким образом, имеющийся набор (коллекция) данных, понимание специфики отношений между этими данными и набора применяемых к данным действий (операций) без уточнения деталей их реализации приводит к появлению важной концепции абстракции (знания того, **ЧТО** можно сделать с соответствующим типом данных, но не **КАК** это делается с точки зрения «неуточненной» реализации (деталей)). В частности, в основе библиотеки языка C++ лежит обширная и достаточно сложная **стандартная библиотека шаблонов (STL)**, являющаяся «библиотекой универсальных компонентов для управления коллекциями данных с помощью эффективных алгоритмов» (Н.Джосаттис. *Стандартная библиотека C++*. Справочное руководство, 2-е изд., «И.Д.Вильямс», 2014). В языках программирования абстракция часто «проявляется посредством пользовательских типов данных; ... для того, чтобы использовать абстрактный тип данных (**АТД**), необходимо понимать его интерфейс, в значительной степени игнорируя детали реализации» (С.Allison. *C&C++ Code Capsules: A Guide for Practitioners*, Prentice Hall PTR, 1998). В своей работе программисты сосредотачивают свое внимание на тех возможностях, которые предоставляет класс, а не на «деталях реализации класса, т.е. они следуют **принципу абстракции данных**: пользовательский код не должен иметь доступа к деталям реализации используемых классов» (W.J.Collins *Data Structures and the Standard Template Library*, McGraw-Hill, 2003). Классы в языке C++ обеспечивают «реализацию АТД (с достаточной степенью сокрытия информации), и любая часть программы, предусматривающая выполнение операции над АТД, сможет это сделать посредством вызова соответствующего метода класса» (M.A.Weiss. *Data Structures and Algorithms Analysis in C++*, 3rd ed., Pearson Educ., 2006).

5 Приблизительный тематический план учебной дисциплины

(учитывая тот факт, что в 2015-2016 учебном году содержание части 2 дисциплины «Алгоритмы и структуры данных» претерпевает некоторые изменения и дается в обновленном формате, то возможны некоторые отклонения от заявленного в плане количества часов, отведенных под обсуждение отдельных тем)

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Практические занятия	
Модуль 3 (44 часа / 22 + 22)		11 января – 26 марта 2016 г. (зачетная неделя : 28 марта – 02 апреля)			



1	Что такое решение задачи? Понятие абстракции, абстрактного типа данных (АТД). Массив как АТД. Спецификация АТД. Типы данных, структуры данных и алгоритмы. Реализация АТД. Указатели. Объектно-ориентированное программирование и С++. Шаблоны, классы. Класс <code>stack</code> как пример контейнера. Рекурсия - вызовы функций (системный стек, запись активации)	19	7	4	8
2	Структуры данных и стандартная библиотека шаблонов (STL) – введение. Классы и объекты в языке С++. Конструкторы, деструктор (особенности и ключевые моменты). Перегрузка операций. Контейнер <code>vector</code> (модель динамического массива). Примеры. Строки. АТД "Строка". Класс <code>string</code> (реализация). Описание строковых классов (типы и операции)	19	5	6	8
3	Слабая типизация (<i>weak typing</i>), указатели на функции. Функции обратного вызова. Функциональные объекты (функторы); применение функторов. Предикатный класс. Динамическое выделение памяти. АТД "Список" (List). Свойства операций над АТД "Список". Связные структуры (списки). Одно- и двусвязные списки. Примеры. Элементы управления памятью. Шаблонные классы <code>array</code> и <code>forward_list</code>	22	6	6	10
4	Массив связанных списков. Представление списка в виде массива. Шаблонный класс <code>list</code> (STL). Пользовательская реализация DLL (шаблонные классы <code>DLLList</code> и <code>Iterator</code>)	18	4	4	10
-	Контрольная работа (КР)	4	-	2	2
• Итого (МОДУЛЬ 3):		82	22	22	38
Модуль 4 (40 часов / 22 + 18)		04 апреля – 18 июня 2016 г. (зачетная неделя: 20 июня – 30 июня)			



5	Ссылочные переменные. Контейнеры (последовательные, ассоциативные, неупорядоченные). Итераторы. Диапазон итераторов (ДИТ). STL контейнеры и итераторы (<i>начало</i>). Примеры	14	4	2	8
6	Стеки, очереди и деки (класс <code>deque</code>). Пользовательские классы стека и очереди (реализация стека и очереди). Приоритетная очередь (куча). Двухуровневая модель дека. Сравнение <code>std::vector</code> и <code>std::deque</code> . Примеры. Стандартные контейнеры и 4 типа итераторов. Адаптеры контейнеров	18	4	6	8
7	Библиотека Boost. Контейнер <code>boost::circular_buffer</code> (кольцо); особенности реализации. STL-совместимые контейнеры; <code>slist</code> , <code>stable_vector</code> и др. Примеры	10	4	-	6
8	Ассоциативные контейнеры – множества/отображения (<code>set</code> , <code>multiset</code> , <code>map</code> , <code>multimap</code>). Примеры использования. Двоичные деревья. Представление двоичных деревьев. АДД "Дерево поиска" – деревья двоичного поиска. 2-3 деревья и 2-3-4 деревья. Примеры	20	6	4	10
9	Хэширование. Разрешение коллизий. Хэш-таблицы в стандартной библиотеке. Качество хэш-функции. Заполнение хэш-таблицы значениями с использованием заданных хэш-функции и подходов	14	4	2	8
-	Практические занятия (дополнительные темы)	6	-	4	2
• Итого (МОДУЛЬ 4):		82	22	18	42
• Итого (МОДУЛИ 3 и 4):		164	44	40	80



6 Формы контроля знаний студентов

Тип контроля	Форма контроля	модули		Параметры
		#3	#4	
Текущий (неделя)	Контрольная работа (КР)	✓ после 17-го марта 2016 г.	✗	Решение предложенного задания на компьютере
	Домашнее задание (ДЗ)	✗	✓ 2-я – 7-я недели модуля	Самостоятельное изучение материала, анализ, реализация и исследование алгоритма(-ов) решения предложенной задачи
Итоговый	Экзамен	✗	✓ после 16-го июня 2016 г.	Тест или решение предложенного задания на компьютере

6.1 Критерии оценки знаний и навыков

Оценки по всем формам контроля выставляются по 10-ти балльной шкале.

Текущий контроль предусматривает одну контрольную работу (КР) (в часы одного из последних практических занятий 3-го модуля) и одно домашнее задание (ДЗ), выполняемое в 4-м модуле (приблизительное время выдачи задания – середина апреля 2016 г.).

Контрольная работа предусматривает программную реализацию на компьютере предложенного задания (алгоритма); выполняется во время часов практического занятия. Оценка за контрольную работу (КР) выставляется с учетом критериев, описанных в п. 9.1.2 учебной программы. Передача контрольной работы с целью повышения оценки не допускается.

Домашнее задание (1 задание, состоящее из одной или нескольких частей) включает анализ решения(-ий) частей задания, реализацию, тестирование и отладку программ(ы). По домашнему заданию оформляется отчет в электронном виде (*файл в формате .pdf*). Домашнее задание загружается студентом в LMS (раздел «Проекты» соответствующей дисциплины) - в установленный и объявляемый заранее срок каждый студент загружает в LMS архив, содержащий полностью оформленный отчет и программу решения предложенного домашнего задания. Оценка за домашнее задание выставляется с учетом полноты выполнения задания и оформления результатов, т.е. при выставлении оценки принимаются во внимание все аспекты выполненной студентом работы.

Итоговый контроль: экзамен в конце 4-го модуля (тест или решение предложенного задания на компьютере).

6.2 Порядок формирования оценок по дисциплине

По всем видам работ оценка выставляется по принятой в НИУ ВШЭ 10-балльной шкале.

О_{модуль2} – промежуточная (для данной части дисциплины) оценка, полученная студентом в конце 2-го модуля (эта оценка проставлена в ведомость по окончании экзамена по дисциплине «Алгоритмы и структуры данных» (часть 1) в конце декабря 2015 г.).



Предусматривается оценивание работ студентов на практических занятиях $O_{\text{пр-класс}}$; для завершения выполнения задания текущего практического занятия дается дополнительное время; результаты (оценки) проверки предыдущего практического занятия должны быть объявлены до начала следующего практического занятия. В оценке учитываются оригинальность, правильность и полнота выполнения выданных для завершения в течение определенного времени заданий по темам практических занятий, стиль оформления кода. Все основные детали и правила организации работы в рамках практических занятий по данной дисциплине будут объяснены преподавателями, проводящими занятия в подгруппах. Оценки за работу на практических занятиях выставляются в рабочую ведомость (электронный вариант). Практические занятия планируется начать со 2-ой недели третьего модуля, т.е. с 18-го января 2016 г.

[Накопление] Накопленная оценка представляет собой взвешенную сумму показанных студентом результатов, а именно: $O_{\text{накопленная}} = 0,15 * O_{\text{модуль2}} + 0,3 * O_{\text{КР}} + 0,25 * O_{\text{пр-класс}} + 0,3 * O_{\text{ДЗ}}$, где $O_{\text{КР}}$ и $O_{\text{ДЗ}}$ – оценки за контрольную работу (КР) 3-го модуля и домашнее задание (ДЗ) 4-го модуля, соответственно. Накопленная оценка $O_{\text{накопленная}}$ не округляется, т.е. используется «as-is» в последующих расчетах.

[Результат] Результирующая оценка (конец четвертого модуля) вычисляется на основе полученной оценки накопления $O_{\text{накопленная}}$ и оценки за экзамен $O_{\text{экзамен}}$ (тест или решение предложенного задания на компьютере) в виде $O_{\text{результат}} = 0,6 * O_{\text{накопленная}} + 0,4 * O_{\text{экзамен}}$. Результирующая оценка $O_{\text{результат}}$ второй части (модули 3и 4) дисциплины – результат арифметического округления.

В случае получения неудовлетворительной оценки и повторной сдачи (т.е. пересдачи) экзамена формула вычисления оценки студента не меняется.

Таблица соответствия оценок по десятибалльной и пятибалльной системам

По десятибалльной шкале	По пятибалльной шкале
1 – неудовлетворительно 2 – очень плохо 3 – плохо	неудовлетворительно – 2
4 – удовлетворительно 5 – весьма удовлетворительно	удовлетворительно – 3
6 – хорошо 7 – очень хорошо	хорошо – 4
8 – почти отлично 9 – отлично 10 – блестяще	отлично – 5

7 Содержание дисциплины (см. стр. 5-7 программы)

7.1 Содержание лекций

Тема 1. Что такое решение задачи? Понятие абстракции, абстрактного типа данных (АТД). Массив как АТД. Спецификация АТД. Типы данных, структуры данных и алгоритмы. Реализация АТД. Указатели. Объектно-ориентированное программирование и С++. Шаблоны, классы. Класс `stack` как пример контейнера. Рекурсия - вызовы функций (системный стек, запись активации)



Литература (источники информации):

1. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. ([4th edition](#) of the book was released in 2014) — **русский перевод книги:** Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
2. Collins W.J. Data Structures and the Standard Template Library, McGraw-Hill HE, 2003, 688 p.
3. Кубенский А.А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++, уч. пособие, СПб.: БХВ-Петербург, 2004, 465 с.
4. Ахо А. В. Хопкрофт Дж.Э., Ульман Дж. Д. Структуры данных и алгоритмы, [уч. пособие](#), ООО «И.Д. Вильямс», 2010, 398 с.

Тема 2. Структуры данных и стандартная библиотека шаблонов (STL) – введение. Классы и объекты в языке C++. Конструкторы, деструктор (особенности и ключевые моменты). Перегрузка операций. Контейнер `vector` (модель динамического массива). Примеры. Строки. АТД "Строка". Класс `string` (реализация). Описание строковых классов (типы и операции)

Литература (источники информации):

1. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. ([4th edition](#) of the book was released in 2014) — **русский перевод книги:** Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
2. Джосаттис Н.М. Стандартная библиотека C++. Справочное руководство, [2-е изд.](#), пер. с англ., ООО «И.Д. Вильямс», 2014, 1136 с. — **английский оригинал книги:** Josuttis N.M. The C++ Standard Library: A Tutorial and Reference, 2nd ed., Addison-Wesley Professional, 2012, 1128 p.
3. Прата С. Язык программирования C++. Лекции и упражнения, [6-е изд.](#), пер. с англ., серия [Landmark](#), ООО «И.Д. Вильямс», 2012, 1248 с.
4. Standard C++ Library Reference, <http://www.cplusplus.com/reference/>

Тема 3. Слабая типизация (*weak typing*), указатели на функции. Функции обратного вызова. Функциональные объекты (функторы); применение функторов. Предикатный класс. Динамическое выделение памяти. АТД "Список" (List). Свойства операций над АТД "Список". Связные структуры (списки). Одно- и двусвязные списки. Примеры. Элементы управления памятью. Шаблонные классы `array` и `forward_list`

Литература (источники информации):

1. Kieras D. Using C++11's Smart Pointers (Course EECS381 "Object-oriented and Advanced Programming), 2014, http://www.umich.edu/~eecs381/handouts/C++11_smart_ptrs.pdf
2. Липпман С.Б., Лажоие Ж., Му Б. Язык программирования C++. Базовый курс, пер. с англ., [5-е изд.](#), ООО «И.Д. Вильямс», 2014, 1120 с.
3. Седжвик Р. Алгоритмы на C++. Анализ, структуры данных, поиск, сортировка, алгоритмы на графах, ООО «И.Д. Вильямс», 2011, 1056 с.
4. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. ([4th edition](#) of the book was released in 2014) — **русский перевод книги:** Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>



5. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных, М.: Финансы и статистика (Инфра-М), 2009, 303 с.
6. Langsam Y., Augenstein M.J., Tenenbaum A.M. Data Structures Using C and C++, 2nd ed., Prentice-Hall, 1996.

Тема 4. Массив связанных списков. Представление списка в виде массива. Шаблонный класс `list` (STL). Пользовательская реализация DLL (шаблонные классы `DLList` и `Iterator`)

Литература (источники информации):

1. Джосаттис Н.М. Стандартная библиотека C++. Справочное руководство, 2-е изд., пер. с англ., ООО «И.Д. Вильямс», 2014, 1136 с. — **английский оригинал книги:** Josuttis N.M. The C++ Standard Library: A Tutorial and Reference, 2nd ed., Addison-Wesley Professional, 2012, 1128 p.
2. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. (4th edition of the book was released in 2014) — **русский перевод книги:** Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
3. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных, М.: Финансы и статистика (Инфра-М), 2009, 303 с.
4. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд., пер. с англ., серия Landmark, ООО «И.Д. Вильямс», 2012, 1248 с.
5. Collins W.J. Data Structures and the Standard Template Library, McGraw-Hill HE, 2003, 688 p.
6. Седжвик Р. Алгоритмы на C++. Анализ, структуры данных, поиск, сортировка, алгоритмы на графах, ООО «И.Д.Вильямс», 2011, 1056 с.

Тема 5. Ссылочные переменные. Контейнеры (последовательные, ассоциативные, неупорядоченные). Итераторы. Диапазон итераторов (ДИТ). STL контейнеры и итераторы (*начало*). Примеры

Литература (источники информации):

1. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. (4th edition of the book was released in 2014) — **русский перевод книги:** Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е изд., пер. с англ., СПб.: ООО «И.Д. Вильямс», 2011., 1296 с. | Кормен Т.Х. Алгоритмы. Вводный курс, пер. с англ., ООО «И.Д.Вильямс», 2014, 208 с.
3. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных, М.: Финансы и статистика (Инфра-М), 2009, 303 с.
4. Седжвик Р. Алгоритмы на C++. Анализ, структуры данных, поиск, сортировка, алгоритмы на графах, ООО «И.Д.Вильямс», 2011, 1056 с.
5. Бабенко М.А., Левин М.В. Введение в теорию алгоритмов и структур данных, 2-е изд. (испр.), М.: МЦНМО, 2014, 140 с.
6. Джосаттис Н.М. Стандартная библиотека C++. Справочное руководство, 2-е изд., пер. с англ., ООО «И.Д. Вильямс», 2014, 1136 с. — **английский оригинал книги:** Josuttis N.M. The C++ Standard Library: A Tutorial and Reference, 2nd ed., Addison-Wesley Professional, 2012, 1128 p.



7. Goodrich M.T., Tamassia R., Mount D.M. Data Structures and Algorithms in C++, 2nd ed., Wiley&Sons, 2011, 735 p.

Тема 6. Стеки, очереди и деки (класс deque). Пользовательские классы стека и очереди (реализация стека и очереди). Приоритетная очередь (куча). Двухуровневая модель дека. Сравнение `std::vector` и `std::deque`. Примеры. Стандартные контейнеры и 4 типа итераторов. Адаптеры контейнеров

Литература (источники информации):

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е изд., пер. с англ., СПб.: ООО «И.Д. Вильямс», 2011., 1296 с. | Кормен Т.Х. Алгоритмы. Вводный курс, пер. с англ., ООО «И.Д.Вильямс», 2014, 208 с.
2. Скиена С. Алгоритмы. Руководство по разработке, 2-е изд., пер. с англ., СПб.: БХВ-Петербург, 2011, 720 с. — **английский оригинал книги:** Skiena S.S. The Algorithm Design Manual, 2nd ed., Springer Verlag, 2008, 728 p.
3. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд., пер. с англ., серия Landmark, ООО «И.Д. Вильямс», 2012, 1248 с.
4. Кубенский А.А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++, уч. пособие, СПб.: БХВ-Петербург, 2004, 465 с.
5. Бабенко М.А., Левин М.В. Введение в теорию алгоритмов и структур данных, 2-е изд. (испр.), М.: МЦНМО, 2014, 140 с.
6. Ахо А. В. Хопкрофт Дж.Э., Ульман Дж. Д. Структуры данных и алгоритмы, уч. пособие, ООО «И.Д. Вильямс», 2010, 398 с.
7. Gilberg R.F., Forouzan B.A. Data Structures. A Pseudocode Approach with C, 2nd ed., Thomson-Course Technology, 2005, 736 с.

Тема 7. Библиотека Boost. Контейнер `boost::circular_buffer` (кольцо); особенности реализации. STL-совместимые контейнеры; `slist`, `stable_vector` и др. Примеры

Литература (источники информации):

1. Boost C++ Libraries - <http://www.boost.org/>
2. Джосаттис Н.М. Стандартная библиотека C++. Справочное руководство, 2-е изд., пер. с англ., ООО «И.Д. Вильямс», 2014, 1136 с. — **английский оригинал книги:** Josuttis N.M. The C++ Standard Library: A Tutorial and Reference, 2nd ed., Addison-Wesley Professional, 2012, 1128 p.
3. Уроки по Boost C++ /Введение/ - <https://www.youtube.com/watch?v=HiaEhvp9nk4>

Тема 8. Ассоциативные контейнеры – множества/отображения (`set`, `multiset`, `map`, `multimap`). Примеры использования. Двоичные деревья. Представление двоичных деревьев. АД "Дерево поиска" – деревья двоичного поиска. 2-3 дерева и 2-3-4 дерева. Примеры

Литература (источники информации):

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е изд., пер. с англ., СПб.: ООО «И.Д. Вильямс», 2011., 1296 с. | Кормен Т.Х. Алгоритмы. Вводный курс, пер. с англ., ООО «И.Д.Вильямс», 2014, 208 с.
2. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. (4th edition of the book was released in 2014) — **русский перевод книги:**



- Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
3. Джосаттис Н.М. Стандартная библиотека C++. Справочное руководство, 2-е изд., пер. с англ., ООО «И.Д. Вильямс», 2014, 1136 с. — **английский оригинал книги:** Josuttis N.M. The C++ Standard Library: A Tutorial and Reference, 2nd ed., Addison-Wesley Professional, 2012, 1128 p.
 4. Ахо А. В. Хопкрофт Дж.Э., Ульман Дж. Д. Структуры данных и алгоритмы, уч. пособие, ООО «И.Д. Вильямс», 2010, 398 с.
 5. Седжвик Р. Алгоритмы на C++. Анализ, структуры данных, поиск, сортировка, алгоритмы на графах, ООО «И.Д.Вильямс», 2011, 1056 с.
 6. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных, М.: Финансы и статистика (Инфра-М), 2009, 303 с.
 7. Collins W.J. Data Structures and the Standard Template Library, McGraw-Hill HE, 2003, 688 p.

Тема 9. Хэширование. Разрешение коллизий. Хэш-таблицы в стандартной библиотеке. Качество хэш-функции. Заполнение хэш-таблицы значениями с использованием заданных хэш-функции и подходов

Литература (источники информации):

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е изд., пер. с англ., СПб.: ООО «И.Д. Вильямс», 2011., 1296 с.
2. Weiss M.A. Data Structures and Algorithm Analysis in C++, 3rd ed., Pearson Education, 2006, 586 p. (4th edition of the book was released in 2014) — **русский перевод книги:** Уайс М.А. Организация структур данных и решение задач, пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
3. Кубенский А.А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++, уч. пособие, СПб.: БХВ-Петербург, 2004, 465 с.
4. Sedgewick R. Algorithms in C++, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, 3rd ed., Addison-Wesley Professional, 1998, 752 p. — **русский перевод книги:** Седжвик Р. Алгоритмы на C++. Анализ, структуры данных, поиск, сортировка, алгоритмы на графах, ООО «И.Д.Вильямс», 2011, 1056 с.

7.2 Содержание практических занятий

Темы практических занятий в рамках дисциплины «Алгоритмы и структуры данных» (часть 2) определяются материалом, рассматриваемым на лекционных занятиях дисциплины и предлагаемым студентам для самостоятельного рассмотрения (изучения). В частности, на практических занятиях могут быть рассмотрены следующие варианты заданий (общее описание тем и направлений, которое может быть дополнено и/или изменено в процессе реализации дисциплины за счет модификации некоторых заданий, появления новых заданий и пр.):

- (1) работа с одно- (*singly*) и двусвязными списками (*doubly linked lists*); двусвязные списки с ограничителями; процедуры *INSERT*, *DELETE*, *UNION* и др. применительно к динамическим структурам; поиск в отсортированном компактном списке,
- (2) представление списка с помощью нескольких массивов, представление с помощью одного массива; выделение и освобождение памяти для элементов двусвязного списка, представленного с помощью нескольких массивов; простой менеджер памяти, основанный на использовании связных списков,
- (3) реализация очереди с помощью двух стеков, анализ времени выполнения операций с элементами очереди; реализация стека с помощью двух очередей, анализ времени вы-



полнения операций с элементами стека; работа со стеком: принятие решений (*backtracking*); обращение (*reverse*) содержимого стека и пр.; кольцевая очередь (*circular queue*),

- (4) двоичные деревья поиска с одинаковыми ключами; задача планирования (запрашиваемого) времени посадки самолетов (отсортированный массив, отсортированный список, дерево двоичного поиска); оценка времени размещения нового элемента в структуре,
- (5) качество хэш-функции; таблицы с прямой адресацией; хэш-таблицы; разрешение коллизий при помощи цепочек; хэширование с открытой адресацией; совершенное (идеальное) хэширование; заполнение хэш-таблицы значениями с использованием заданной хэш-функции и различных подходов (*probing*).

8 Образовательные технологии

Работа на практических занятиях (ПР) предполагает *самостоятельную* разработку программ. Языком программной реализации алгоритмов является язык программирования C++ — на лекционных и практических занятиях дисциплины рассматриваются элементы языка, необходимые для представления тем лекций, и стандартной библиотеки в объеме, достаточном для выполнения предлагаемых практических заданий. Домашнее задание (ДЗ) предполагает *самостоятельное изучение* нового материала, анализ путей решения предлагаемых заданий, реализацию, тестирование и отладку программ(ы), связанную с использованием определенных алгоритмов и соответствующих структур данных.

9 Оценочные средства для текущего контроля и аттестации студента

9.1 Тематика заданий текущего контроля

9.1.1 Домашнее задание

Предусматривается подробный анализ, разработка алгоритма решения и его программная реализация применительно к предложенной проблеме (задаче), связанной с рассматриваемым в рамках данной дисциплины материалом и темами, ориентированными на самостоятельное изучение.

9.1.2. Контрольная работа

Примерное задание (шаблон) описано в программе дисциплины «Алгоритмы и структуры данных» (часть 1). Там же приведено и краткое описание подхода к выставлению оценки за контрольную работу (*оставлено без изменений* — см. **стр.13** рабочей программы дисциплины «Алгоритмы и структуры данных» (**часть 1**) 2015-2016 учебного года), а именно:

- 1) Алгоритм реализован верно — **4 балла** (из 10 возможных),
- 2) Верно выполняется тестовый пример, показанный в описании задания — **5 баллов**,
- 3) Тестовый пример выполняется не на всех входных данных — (+ до 2 баллов) - **6-7 баллов**.
- 4) Тестовый пример выполняется на всех входных данных (+ до 2 баллов) - **8-9 баллов**.

Работа выполнена самостоятельно менее, чем за 60 минут — +1 балл к оценке, полученной в соответствии с пп. 1)-4).

9.2 Отдельные примерные вопросы для оценки качества освоения дисциплины

1. Как бы вы определили понятие «структура данных»?
2. Как бы вы определили понятие «абстракция данных»? Что такое «абстракция» и в чем состоит ее важность (значимость). Приведите пример абстрактного типа данных. Сформулируйте и кратко прокомментируйте преимущества использования АД



3. Что собой представляет динамическая структура данных?
4. Определите АД для символьных строк (АД должен включать типичные функции, которые применяются к строкам; каждую функцию определите в терминах входных и выходных данных). Предложите две разные реализации строкового типа
5. Что такое дек? Что такое стек? Объясните, чем отличается стек от очереди. Как выглядит пользовательский класс стека, очереди, очереди с приоритетом (STL)?
6. Объясните отличие класса (`class`) от структуры (`struct`)
7. Какие из <предложенных> структур данных являются индексруемыми структурами?
8. Что такое связный список? Как можно представить разреженную матрицу (*sparse matrix*) с использованием связного списка? Как можно эффективно представить разреженную матрицу с использованием массива? Что можно сказать о сложности хранения (*space complexity*) такого подхода?
9. Приведите примеры использования векторов, деков, списков, множеств (мульти-множеств) (STL)
10. Предположим, что текущая конфигурация очереди имеет вид: a b c d. Сколько понадобится операций извлечения/размещения для получения конфигурации d c b a?
11. Как организовать поиск заданного элемента в односвязном списке (SLL)?
12. Что собой представляет дерево бинарного поиска (*binary search tree*)?
13. Что собой представляют классы `shared_ptr` и `unique_ptr` (типы интеллектуальных указателей)?
14. Какой алгоритм сортировки считается самым быстрым?
15. Какие вы знаете типы контейнеров (STL)?
16. Какую работу выполняет хэш-функция?
17. Предположим, что задана функция $f(x) = x \% 7$ и используется линейное исследование (*linear probing*) для размещения ключей 37,38,72,48,98,11,56 в таблице с индексацией элементов от 0 до 6. На какой позиции в таблице будет размещен элемент 11?

10 Учебно-методическое и информационное обеспечение дисциплины

10.1 Базовый учебник — отсутствует

10.2 Основная литература:

1. (на англ. языке) Weiss M.A. Data Structures and Algorithm Analysis in C++, 4th ed., Pearson Education, 2013, 654 p. — **русский перевод книги**: Уайс М.А. Организация структур данных и решение задач (3-е издание), пер. с англ., М.: Эком, 2009, 896 с. | М.А.Weiss Homepage - <http://users.cis.fiu.edu/~weiss/>
2. Джосаттис Н.М. Стандартная библиотека C++. Справочное руководство, 2-е изд., пер. с англ., ООО «И.Д. Вильямс», 2014, 1136 с. — **английский оригинал книги**: Josuttis N.M. The C++ Standard Library: A Tutorial and Reference, 2nd ed., Addison-Wesley Professional, 2012, 1128 p.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е изд., пер. с англ., СПб.: ООО «И.Д. Вильямс», 2011., 1296 с. | Кормен Т.Х. Алгоритмы. Вводный курс, пер. с англ., ООО «И.Д.Вильямс», 2014, 208 с.
4. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд., пер. с англ., серия Landmark, ООО «И.Д. Вильямс», 2012, 1248 с.
5. Седжвик Р. Алгоритмы на C++. Анализ, структуры данных, поиск, сортировка, алгоритмы на графах, ООО «И.Д.Вильямс», 2011, 1056 с. (2 книги в одной) | Sedgewick R. Algorithms in C++, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, 3rd ed., Addison-Wesley Professional, 1998, 752 p.



6. (на англ. языке) Sedgewick R., Wayne K. Algorithms, 4th ed., Addison-Wesley Prof., 2011, 990 p.
7. (на англ. языке) Goodrich M.T., Tamassia R., Mount D.M. Data Structures and Algorithms in C++, 2nd ed., Wiley&Sons, 2011, 735 p.

10.3 Дополнительная литература и источники

1. Скиена С. Алгоритмы. Руководство по разработке, 2-е изд., пер. с англ., СПб.: БХВ-Петербург, 2011, 720 с. — **английский оригинал книги**: Skiena S.S. The Algorithm Design Manual, 2nd ed., Springer Verlag, 2008, 728 p.
2. Макконелл Дж. Основы современных алгоритмов, 2-е доп. изд., пер. с англ., М.: Техносфера, 2006, 368 с.
3. Ахо А. В. Хопкрофт Дж.Э., Ульман Дж. Д. Структуры данных и алгоритмы, [уч. пособие](#), ООО «И.Д. Вильямс», 2010, 398 с.
4. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных, М.: Финансы и статистика (Инфра-М), 2009, 303 с.
5. Бабенко М.А., Левин М.В. Введение в теорию алгоритмов и структур данных, 2-е изд. (испр.), М.: МЦНМО, 2014, 140 с.
6. Липпман С.Б., Лажоие Ж., Му Б. Язык программирования C++. Базовый курс, пер. с англ., [5-е изд.](#), ООО «И.Д. Вильямс», 2014, 1120 с.
7. Кубенский А.А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++, уч. пособие, СПб.: БХВ-Петербург, 2004, 465 с.
8. (на англ. языке) Langsam Y., Augenstein M.J., Tenenbaum A.M. Data Structures Using C and C++, 2nd ed., Prentice-Hall, 1996.
9. (на англ. языке) Gilberg R.F., Forouzan B.A. Data Structures. A Pseudocode Approach with C, 2nd ed., Thomson-Course Technology, 2005, 736 с.
10. (на англ. языке) Collins W.J. Data Structures and the Standard Template Library, McGraw-Hill HE, 2003, 688 p.
11. (на англ. языке) Forouzan B.A., Mosharraf F. Foundations of Computer Science, 2nd ed., Thomson Learning (Cengage), 2008, 624 p.

10.4 Справочные материалы, словари, энциклопедии, инф. сайты и доп. книги

1. (на англ. языке) Collected Algorithms (ACM), <http://www.acm.org/calgo>
2. TutorialPoint: C++ STL Tutorial, http://www.tutorialspoint.com/cplusplus/cpp_stl_tutorial.htm
3. Контейнеры STL (MSDN), <http://msdn.microsoft.com/ru-ru/library/1fe2x6kt.aspx>
4. (на русск. языке) Справка по C++, <http://ru.cppreference.com/w/> | (на англ. языке) C++ Reference, <http://en.cppreference.com/w/>
5. (на англ. языке) Standard C++ Library Reference, <http://www.cplusplus.com/reference/>
6. (на англ. языке) Shaffer C.A. Data Structures and Algorithm Analysis (C++ version/3.2), Virginia Tech, 2010-13, people.cs.vt.edu/shaffer/Book/C++3elatest.pdf
7. Using C++11's Smart Pointers (Course EECS381 "Object-oriented and Advanced Programming Handouts, UM), 2014, http://www.umich.edu/~eecs381/handouts/C++11_smart_ptrs.pdf
8. (на англ. языке) [Meyers S.](#) Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library, Addison-Wesley Prof., 2001, 286 p.
9. (на англ. языке) Raykar M. Implementing a simple smart pointer in C++, CodeProject, 2006, <http://www.codeproject.com/Articles/15351/Implementing-a-simple-smart-pointer-in-c>



10.5 Программные средства

Практические занятия проводятся в компьютерных классах с выходом в Интернет и доступом к ресурсам электронной библиотеки (<http://library.hse.ru/e-resources/e-resources.htm>) НИУ ВШЭ. Предусматривается наличие у каждого студента рабочего места. Практическая работа ориентирована на использование современ. интегрированных инструментальных сред разработки.

••• **Примечание:** для самостоятельной работы (установки на персональных компьютерах) студенты могут использовать другие C++ IDE – например, Code::Blocks 15.12 (www.codeblocks.org) / RC1 (по состоянию на 29.12.15), JetBrains CLion 1.2.x (<https://www.jetbrains.com/clion/>), Orwell Dev-C++ 5.11 (<http://sourceforge.net/projects/orwelldevcpp/>), CodeLite 9.0 (<http://codelite.org>), Qt (<http://www.qt.io>) и др. (напр., gcc 4.9.x или более поздние релизы).

Используемое программное обеспечение:

1. Microsoft Office Professional 2007-2010 (или, более поздние версии),
2. MS Visual Studio 2012 (2013 или более поздние версии).

10.6 Дистанционная поддержка дисциплины

Дистанционная поддержка дисциплины обеспечивается использованием LMS. В разделе дисциплины «Анализ и структуры данных» (часть 2) размещаются дополнительные материалы, связанные с лекциями, практическими занятиями, материалы для самоподготовки, проекты, оценки текущего и итогового контроля.

11 Материально-техническое обеспечение дисциплины

Проектор (лекционные и практические занятия), классы для практических занятий, оснащенные компьютерами с установленной инструментальной средой разработки MS Visual Studio 2012/2013.