

УТВЕРЖДАЮ
 Декан факультета

_____ Матвеев П.В.

« ____ » _____ 20__

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Направление/специальность подготовки	24.05.06 Системы управления летательными аппаратами
Специализация/профиль/программа подготовки	Системы управления ракет
Уровень высшего образования	Специалитет
Форма обучения	Очная
Факультет	И Информационных и управляющих систем
Выпускающая кафедра	И9 СИСТЕМ УПРАВЛЕНИЯ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Кафедра-разработчик рабочей программы	О7 Информационные системы и программная инженерия

КУРС	СЕМЕСТР	ОБЩАЯ ТРУДОЁМКОСТЬ (ЗАЧЕТНЫХ ЕДИНИЦ)	ЧАСЫ (по наличию видов занятий)									ВИД ПРОМЕЖУТОЧНОГО КОНТРОЛЯ
			ОБЩАЯ ТРУДОЁМКОСТЬ	АУДИТОРНЫЕ ЗАНЯТИЯ				САМОСТОЯТЕЛЬНАЯ РАБОТА				
				ВСЕГО	ЛЕКЦИИ	ЛАБОРАТОРНЫЙ ПРАКТИКУМ	ПРАКТИЧЕСКИЕ ЗАНЯТИЯ	ВСЕГО	КУРСОВОЙ ПРОЕКТ	КУРСОВАЯ РАБОТА	ДРУГИЕ ВИДЫ САМОСТ. РАБОТЫ	
2	3	4	144	51	17	0	34	93	0	0	93	диф. зач.

ЛИСТ СОГЛАСОВАНИЯ

РАБОЧАЯ ПРОГРАММА СОСТАВЛЕНА В СООТВЕТСТВИИ С ТРЕБОВАНИЯМИ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ВЫСШЕГО ОБРАЗОВАНИЯ (ФГОС ВО)

24.05.06 Системы управления летательными аппаратами

год набора группы: 2025

Программу составил:

Кафедра О7 Информационные системы и программная инженерия
Макаренко Александр Александрович, к.т.н., доцент, доцент

Программа рассмотрена
на заседании кафедры-разработчика
рабочей программы **О7 Информационные системы и программная инженерия**

Заведующий кафедрой Семенова Е.Г., д.т.н., проф.

Программа рассмотрена
на заседании выпускающей кафедры

И9 СИСТЕМ УПРАВЛЕНИЯ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Заведующий кафедрой Сырцев А.Н., д.воен.н., снс

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Разделы рабочей программы

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО
3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ
4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Приложения к рабочей программе дисциплины

- Приложение 1. Аннотация рабочей программы
- Приложение 2. Технологии и формы обучения
- Приложение 3. Фонды оценочных средств

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является формирование следующих компетенций:

ПК-5 — Способен разрабатывать алгоритмы и программное обеспечение для системы управления летательным аппаратом и математических моделей систем управления

ОПК-2 — Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

ОПК-9 — Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

Формированию компетенций служит достижение следующих результатов образования:

ПК-5

знания:

Знание языка программирования и сфер его применения;;

умения:

Умение выполнять расчет математических выражений с помощью языка программирования;;

навыки:

Владение инструментами создания среды виртуальной реальности;.

ОПК-2

знания:

уметь составлять алгоритмы решения различных задач на компьютере и использовать методы структурного и объектно-ориентированного программирования, познакомиться с динамическими структурами данных, научиться создавать абстрактные типы данных и решать задачи с использованием списков, стеков и очередей

реализовывать и проектировать поставленные задачи с помощью языка программирования C++, выдвигать требования к структуре хранения данных и способам их обработки, оформлять и анализировать полученные в ходе расчетов результаты

применять полученные знания в дальнейшем при решении задач, связанных с проектированием систем управления в области приборостроения, использовать в расчетах современные компьютерных технологий языков программирования высокого уровня;

умения:

владеть методами разработки алгоритмов, использовать инструменты системного анализа в современной программно-технической среде в различных операционных системах;

навыки:

разрабатывать модели и отлаживать алгоритмы с использованием современных технологий программирования

работать в качестве пользователя персонального компьютера, использовать языки и системы программирования для решения профессиональных задач, разрабатывать программные приложения с обоснованием методов обработки и хранения данных на уровне сложных линейных связанных структур.

ОПК-9

знания:

уметь составлять алгоритмы решения различных задач на компьютере и использовать методы структурного и объектно-ориентированного программирования, познакомиться с динамическими структурами данных, научиться создавать абстрактные типы данных и решать задачи с использованием списков, стеков и очередей;

умения:

владеть методами разработки алгоритмов, использовать инструменты системного анализа в современной программно-технической среде в различных операционных системах;

навыки:

разрабатывать модели и отлаживать алгоритмы с использованием современных технологий программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина **ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *24.05.06 Системы управления летательными аппаратами*.

Содержание дисциплины является логическим продолжением дисциплин: **ВВЕДЕНИЕ В ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ**.

Содержание дисциплины является основой для освоения дисциплин: **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ**.

Предварительные компетенции, сформированные у обучающегося до начала изучения дисциплины:

- ОПК-2 — Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
- ОПК-9 — Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения
- ПК-5 — Способен разрабатывать алгоритмы и программное обеспечение для системы управления летательным аппаратом и математических моделей систем управления
- ПК-93 — Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов
- ПК-94 — Способен к управлению информацией и данными, поиску источников информации и данных, восприятию, анализу, запоминанию и передаче информации с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 з.е., 144 ч.

3.1. Содержание (дидактика) дисциплины

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %		
				ВСЕГО	Лекции	Практические занятия		ПК-5	ОПК-2	ОПК-9
2	3	Раздел 1. Функции. Функции: фактические и формальные параметры, прототип, вызов функции, локальные и глобальные объекты, механизм передачи параметров, работа с указателями на объекты. Указатель на функцию: классификация указателей на функцию, объявление, определение инициализация, вызов функции через указатель, указатель на функцию, как параметр другой функции, вычисление интегралов с помощью указателя на функцию разными методами. Функции с переменным количеством параметров: описание функции с переменным количеством параметров, вызов функции, механизмы передачи параметров. Рекурсивные функции: описание рекурсивной функции, вызов функции, прямая и косвенная рекурсии, «дно» рекурсии, вычисление факториалов и степени с помощью рекурсии. Функция <code>main()</code> : параметры функции <code>main()</code> , назначение и применение параметров, командная строка, операционная система.	31	9	3	6	22	25	25	30
2	3	Раздел 2. Файлы. Символьная информация: объявление, определение инициализация символьных объектов, функции работы со строками, указатели на строки, динамическое выделение памяти для символьных объектов. Текстовые файлы: указатель на файл, функция <code>fopen()</code> , режимы открытия файла, функции чтения и записи в файлы, функции для работы с файлами. Бинарные файлы: указатель на файл, функция <code>fopen()</code> , режимы открытия файла, функции чтения и записи в файлы, функции для работы с файлами, примеры создания бинарного файла.	32	9	3	6	23	25	25	25
2	3	Раздел 3. Структурные типы данных. Структуры: создание структурного типа данных <code>struct</code> , объединение <code>union</code> , битовые поля, объявление, определение, инициализация объекта и указателя структурных типов. Классы: понятие класса, структура и определение, объект класса, указатель на класс, конструктор, деструктор, поля класса, методы класса, дружественные функции, указатель <code>this</code> , динамическое выделение памяти под объекты класса.	33	10	4	6	23	25	25	25
2	3	Раздел 4. Списки. Односвязные списки: линейный односвязный список, правила выделения памяти и обработки элементов списка, информационные поля, поля связи, добавление, удаление, сортировка и поиск элементов в списке. Двусвязные списки: линейный двусвязный список, правила выделения памяти и обработки элементов списка, информационные поля, поля связи, добавление, удаление, сортировка и поиск элементов в списке.	48	23	7	16	25	25	25	20
Всего за 3 семестр			144	51	17	34	93	100	100	100
Всего по дисциплине			144	51	17	34	93	100	100	100

3.2. Аудиторный практикум

№ п/п	Номер и наименование раздела дисциплины	Тема практического занятия	Объем, ауд. часов
1	Раздел 1. Функции.	Функции: функции, фактические и формальные параметры, прототип, вызов функции, механизм передачи параметров, указатель на функцию, вычисление интегралов с помощью указателя на функцию разными методами, рекурсивные функции, функции с переменным количеством параметров, функция <code>main()</code>	6
2	Раздел 2. Файлы.	Символьная информация: объявление символьных объектов, функции работы со строками, указатели на строки	2
3		Файлы: текстовые файлы, указатель на файл, открытие текстового файла, функции для работы с текстовыми файлами, бинарные файлы, указатель на файл, открытие бинарного файла, функции для работы с бинарными файлами	4
4	Раздел 3. Структурные типы данных.	Структурные типы данных: классы, структурные типы данных, объект класса, указатель на объект, поля класса и методы класса	6
5	Раздел 4. Списки.	Списки: линейные списки, обработка элементов списка, динамическое выделение памяти для элементов списка, классификация списков и	16

	правила их обработки	
Всего за 3 семестр		34

3.3. Самостоятельная работа студента (СРС)

№ п/п	Номер и наименование раздела дисциплины	Содержание учебного задания	Объем, часов
1	Раздел 1. Функции.	изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	13
2		подготовка к практическим заданиям: изучение раздела 1 и оформление отчетов к практическим заданиям 1	9
3	Раздел 2. Файлы.	изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	13
4		подготовка к практическим заданиям: изучение раздела 2 и оформление отчетов к практическим заданиям 2	10
5	Раздел 3. Структурные типы данных.	изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	13
6		подготовка к практическим заданиям: изучение раздела 3 и оформление отчетов к практическим заданиям 3	10
7	Раздел 4. Списки.	изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	15
8		подготовка к практическим заданиям: изучение раздела 4 и оформление отчетов к практическим заданиям 4	10
Всего за 3 семестр			93

4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

СЕМЕСТР	НЕДЕЛИ СЕМЕСТРА																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3				Отч. по ПЗ, ИПЗ		ДР		ИПЗ, Отч. по ПЗ		ДР		ИПЗ, Отч. по ПЗ			Отч. по ПЗ, ИПЗ	ДР	диф. зач.

Условные обозначения:

- ДР – диагностическая работа;
- Отч. по ПЗ – отчет по практическому заданию;
- ИПЗ – индивидуальное практическое задание;
- диф. зач. – дифференцированный зачет.

Текущий контроль успеваемости студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- отчет по практическому заданию;
- индивидуальное практическое задание.

Промежуточная аттестация проводится в формах:

- дифференцированный зачет.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Основная литература по дисциплине:

1. . Основы программирования на языке Си. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2006, эл. рес.
2. Б. П. Арсеньев, О. А. Решетова, И. И. Рыкова. Основы языков программирования С и С++. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2001, 168 экз.
3. И. Г. Головин, И. А. Волкова . . Языки и методы программирования. М.: Академия, 2016, 50 экз.
4. Н. И. Парфилова, А. В. Пруцков, А. Н. Пылькин. . Информатика и программирование. Основы информатики. М.: Академия, 2012, 15 экз.
5. О. А. Палехова. . Основы программирования на языке Си. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2016, 390 экз.
6. О. А. Палехова. . Основы программирования на языке Си. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2016, эл. рес.
7. О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014, 60 экз.
8. С. А. Орлов. . Теория и практика языков программирования. СПб.: Питер, 2013, 30 экз.

5.2. Дополнительная литература по дисциплине:

не требуется.

5.3. Периодические издания:

1. Прикладная информатика.

5.4. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины, электронные библиотечные системы:

1. <https://urait.ru/> — Главная – Образовательная платформа Юрайт. Для вузов и ссузов.;
2. <https://library.voenmeh.ru/> — Фундаментальная библиотека БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова.

Современные профессиональные базы данных:

1. <https://rusneb.ru> – Национальная электронная библиотека (НЭБ);
2. <https://cyberleninka.ru/> - Научная электронная библиотека «Киберленинка»;
- <http://www.rfbr.ru/rffi/ru/library> - Полнотекстовая электронная библиотека Российского фонда фундаментальных исследований.

Информационные справочные системы:

1. Техэксперт – Информационный портал технического регулирования: Нормы, правила, стандарты РФ;
2. http://library.voenmeh.ru/jirbis2/index.php?option=com_irbis&view=irbis&Itemid=457 - БД ГОСТов собственной генерации БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова;
3. <http://www.consultant.ru/>- КонсультантПлюс- информационный портал правовой информации.

5.5. Программное обеспечение:

1. Bloodshed Dev-C++;
2. Code::Blocks.

5.6. Информационные технологии:

взаимодействие с обучающимися посредством ЭИОС Moodle БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Лекционные занятия:

специализированные требования по оборудованию отсутствуют; аудитория с посадочными местами по количеству студентов; доска.

6.2. Практические занятия:

1. Проектор;
2. Аудитория с числом посадочных мест не меньше количества обучающихся;
3. Bloodshed Dev-C++;
4. Code::Blocks.

6.3. Прочее:

1. рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
2. рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

Аннотация рабочей программы

Дисциплина **ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *24.05.06 Системы управления летательными аппаратами*. Дисциплина реализуется на факультете *О Естественнoнаучный* БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова кафедрой *О7 Информационные системы и программная инженерия*.

Дисциплина нацелена на формирование *компетенций*:

ПК-5 Способен разрабатывать алгоритмы и программное обеспечение для системы управления летательным аппаратом и математических моделей систем управления;

ОПК-2 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности;

ОПК-9 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

Содержание дисциплины охватывает круг вопросов, связанных с изучением современных средств, предназначенных для формирования у студентов глубоких теоретических знаний в области управления, хранения и обработки данных, а также практических навыков по проектированию и реализации эффективных систем хранения и обработки данных на основе полученных знаний.

Программой дисциплины предусмотрены следующие **виды контроля**:

Текущий контроль успеваемости студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- отчет по практическому заданию;
- индивидуальное практическое задание.

Промежуточная аттестация проводится в формах:

- дифференцированный зачет.

Общая трудоемкость освоения дисциплины составляет **4 з.е., 144 ч.** Программой дисциплины предусмотрены лекционные занятия (**17 ч.**), практические занятия (**34 ч.**), самостоятельная работа студента (**93 ч.**).

ТЕХНОЛОГИИ И ФОРМЫ ОБУЧЕНИЯ

Рекомендации по освоению дисциплины для студента

Трудоемкость освоения дисциплины составляет 144 ч., из них 51 ч. аудиторных занятий, и 93 ч., отведенных на самостоятельную работу студента.

Рекомендации по распределению учебного времени по видам самостоятельной работы и разделам дисциплины приведены в таблице.

Контроль освоения дисциплины производится в соответствии с Положением о текущем, рубежном контроле успеваемости и промежуточной аттестации обучающихся.

Формы контроля и критерии оценивания приведены в приложении 3 к Рабочей программе.

Наименование работы	Рекомендуемая литература	Трудоемкость, час.
Раздел 1. Функции.		
изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	О. А. Палехова. . Основы программирования на языке Си: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2016 (7)	13
подготовка к практическим заданиям: изучение раздела 1 и оформление отчетов к практическим заданиям 1	. Основы программирования на языке Си: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2006 (7) Б. П. Арсеньев, О. А. Решетова, И. И. Рыкова. Основы языков программирования С и С++: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2001 (7)	9
Итого по разделу 1		22
Раздел 2. Файлы.		
изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	. Основы программирования на языке Си: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2006 (8) О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1)	13
подготовка к практическим заданиям: изучение раздела 2 и оформление отчетов к практическим заданиям 2	О. А. Палехова. . Основы программирования на языке Си: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2016 (8) Б. П. Арсеньев, О. А. Решетова, И. И. Рыкова. Основы языков программирования С и С++: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2001 (8)	10
Итого по разделу 2		23
Раздел 3. Структурные типы данных.		
изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (2)	13
подготовка к практическим заданиям: изучение раздела 3 и оформление отчетов к практическим заданиям 3	Н. И. Парфилова, А. В. Пруцков, А. Н. Пылькин. . Информатика и программирование. Основы информатики: М.: Академия, 2012 (2) И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (2) С. А. Орлов. . Теория и практика языков программирования: СПб.: Питер, 2013 (2)	10
Итого по разделу 3		23

Раздел 4. Списки.		
изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (3)	15
подготовка к практическим заданиям: изучение раздела 4 и оформление отчетов к практическим заданиям 4	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (3) Н. И. Парфилова, А. В. Пруцков, А. Н. Пылькин. . Информатика и программирование. Основы информатики: М.: Академия, 2012 (3) С. А. Орлов. . Теория и практика языков программирования: СПб.: Питер, 2013 (3)	10
Итого по разделу 4		25

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств, позволяющие оценить результаты обучения по данной дисциплине, включают в себя:

- диагностическая работа
- отчет по практическому заданию;
- индивидуальное практическое задание;
- дифференцированный зачет.

Критерии оценивания

Диагностическая работа

Диагностическая работа проводится в форме теста в ЭИОС Moodle:

- при правильном ответе менее чем на 60% вопросов - не аттестация;
- при правильном ответе на 60% вопросов и более - аттестация.

Отчет по практическому заданию

К каждому индивидуальному практическому заданию оформляется отчет в электронном виде, состоящий из титульного листа установленного образца, вариативной части задания и результатов работы программы.

Отчет оформляется согласно ГОСТ 7.32-2017.

Индивидуальное практическое задание

Перечень индивидуальных заданий:

1. Функции
2. Строки
3. Файлы
4. Списки

Индивидуальное задание считается выполненным и защищенным успешно при условии:

- наличия программного приложения, реализующего поставленную задачу;
- наличия отчета;
- защиты индивидуального задания в форме ответов на вопросы по заданию и ответов на контрольные вопросы, приведенные к комплекту типовых заданий по каждому разделу.

Допуск к выполнению ИПЗ не предусмотрен.

Подробные критерии оценивания ИПЗ указаны в технологической карте дисциплины, размещённой в курсе в ЭИОС.

Дифференцированный зачет

График контрольных мероприятий предусматривает выполнение студентом четырёх индивидуальных заданий. Дифференцированный зачет выставляется по сумме результатов контрольных мероприятий, проводимых в течение семестра. Максимальная сумма баллов за семестр – 100 баллов. Набранная итоговая сумма баллов пересчитывается в оценку по следующей схеме:

- 85 – 100 баллов – отлично;
- 75 – 84 балла – хорошо;
- 51 – 74 баллов – удовлетворительно;
- меньше 51 – не зачтено.

В случае несогласия студента с оценкой согласно набранным баллам, может быть проведён устный зачёт, вопросы к которому располагаются в УМК дисциплины. В этом случае дифференцированный зачёт проходит в форме ответов на два вопроса из перечня и решения задачи.

Зачтено-удовлетворительно - студент владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускаются ошибки по существу вопросов. Студент способен решать лишь наиболее легкие задачи, владеет только обязательным минимумом знаний.

Зачтено-хорошо - студент владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах дает полноценные ответы на вопросы; не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах.

Зачтено-отлично - студент владеет знаниями предмета в полном объеме учебной программы,

достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивает при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное; устанавливать причинно-следственные связи; четко формирует ответы, решает сложные задачи.

Оценка за зачёт ставится с учётом работы в семестре или корректных и полных ответов на вопросы и решения задачи.

Паспорт фонда оценочных средств

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %			НАИМЕНОВАНИЕ ОЦЕНОЧНОГО СРЕДСТВА
				ВСЕГО	Лекции	Практические занятия		ПК-5	ОПК-2	ОПК-9	
2	3	Раздел 1. Функции.	31	9	3	6	22	25	25	30	Отчет по практическому заданию, Индивидуальное практическое задание
2	3	Раздел 2. Файлы.	32	9	3	6	23	25	25	25	Отчет по практическому заданию, Индивидуальное практическое задание
2	3	Раздел 3. Структурные типы данных.	33	10	4	6	23	25	25	25	Отчет по практическому заданию, Индивидуальное практическое задание
2	3	Раздел 4. Списки.	48	23	7	16	25	25	25	20	Отчет по практическому заданию, Индивидуальное практическое задание
Всего за 3 семестр			144	51	17	34	93	100	100	100	
Всего по дисциплине			144	51	17	34	93	100	100	100	

Оценочные материалы по дисциплине ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

ПК-5 - Способен разрабатывать алгоритмы и программное обеспечение для системы управления летательным аппаратом и математических моделей систем управления

- № 1 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа
Какая операция в односвязном списке выполняется за время $O(1)$ (константное время)?
1. Удаление элемента из середины списка.
 2. Вставка элемента в начало списка.
 3. Поиск элемента по значению.
 4. Сортировка списка пузырьком.
- № 2 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа
Какое преимущество двусвязного списка перед односвязным при удалении элемента?
1. Не требует хранения указателя на следующий элемент.
 2. Позволяет удалить элемент за $O(1)$ без предварительного поиска.
 3. Упрощает удаление элемента, если известен только его адрес (без перебора предыдущих).
 4. Всегда работает быстрее односвязного списка.
- № 3 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа
В каком случае двусвязный список предпочтительнее односвязного?
1. Когда требуется частая вставка элементов в начало.
 2. Когда нужен быстрый доступ к произвольному элементу по индексу.
 3. Когда часто выполняются операции удаления/вставки в середину списка.
 4. Когда важно минимизировать расход памяти.
- № 4 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов
Какие утверждения верны относительно формальных и фактических параметров функций в C++?
1. Формальные параметры — это переменные, объявленные в прототипе функции.
 2. Фактические параметры — это значения, передаваемые в функцию при вызове.
 3. Локальные объекты функции видны в других функциях, если объявлены как `static`.
 4. Глобальные объекты можно изменять внутри любой функции без их передачи в качестве параметров.
- № 5 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов
Какие операции допустимы с указателем на функцию в C++?
1. Присваивание другой функции с совпадающей сигнатурой.
 2. Арифметические операции (например, ++ или + n).
 3. Передача в другую функцию как параметр.
 4. Использование для вычисления интеграла методом прямоугольников.
- № 6 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов
Какие утверждения верны для рекурсивных функций и функции `main`?
1. Рекурсия требует обязательного условия выхода («дно рекурсии»).
 2. Параметры `main(int argc, char* argv[])` получают данные из командной строки.
 3. Косвенная рекурсия — это когда функция A вызывает B, а B вызывает A.
 4. `main()` нельзя вызывать рекурсивно.

- № 7 Прочитайте текст и запишите развернутый обоснованный ответ
Почему строки в стиле C обязательно завершаются нуль-терминатором (\0)? Что произойдет, если его не добавить?
- № 8 Прочитайте текст и запишите развернутый обоснованный ответ
Объясните разницу между: Строками в стиле C (char[], const char*) и классом std::string.
Как хранится строка "Hello" в памяти для каждого случая?
- № 9 Прочитайте текст и установите соответствие
Установите соответствие между терминами, связанными с классами в C++, и их определениями.

1	Инкапсуляция	А	Механизм, позволяющий одному классу наследовать свойства и методы другого
2	Наследование	Б	Скрытие внутренней реализации класса и предоставление строго определённого интерфейса
3	Полиморфизм	В	Возможность использовать один и тот же интерфейс для работы с разными типами данных
4	Конструктор	Г	Специальный метод класса, вызываемый при создании объекта
		Д	Специальный метод класса, вызываемый при уничтожении объекта

- № 10 Прочитайте текст и установите соответствие
Установите соответствие между модификаторами доступа в классах C++ и их свойствами.

1	public	А	Члены класса доступны только внутри самого класса и его дружественных функций.
2	private	Б	Члены класса доступны из любого места программы.
3	protected	В	Члены класса доступны внутри класса,

его наследников и дружественных функций.
Функция или класс, имеющие доступ к private и protected членам другого класса.

№ 11 Прочитайте текст и установите последовательность

Установите правильный порядок действий при безопасном чтении из файла.

1. Открыть файл в режиме чтения (например, `std::ifstream`).
2. Проверить, открылся ли файл (`is_open()`).
3. Проверить, достигнут ли конец файла (`eof()`).
4. Считать данные (используя `>>`, `getline()` или `read()`).
5. Закрыть файл.

№ 12 Прочитайте текст и установите последовательность

Установите правильный порядок действий при объявлении, определении и вызове функции.

1. Объявление функции (прототип в заголовочном файле или перед `main`).
2. Определение функции (реализация с телом функции).
3. Передача аргументов в функцию (если они есть).
4. Выполнение кода внутри функции.
5. Возврат значения (если функция не `void`).
6. Использование результата (если функция возвращает значение).

ОПК-2 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

№ 1 Прочитайте текст и установите соответствие

Для термина в области программирования в правом столбце, подберите определение из левого столбца.

- | | | |
|----|--|---|
| 1. | Теоретическая и практическая деятельность, связанная с созданием компьютерных программ | А. Программирование |
| 2. | Совокупность методов и средств для сокращения стоимости и повышения качества создания программного обеспечения | Б. проектирование программного продукта |
| 3. | Выбор архитектуры программного обеспечения; типа пользовательского интерфейса; структурного или объектного подхода к разработке; языка и среды для создания программ | В. Технология программирования |
| | | Г. Тестирование программного продукта |

№ 2 Прочитайте текст и установите соответствие

Поставьте в соответствие фрагменты программы

- | | | |
|----|--------------------------|----------------------------------|
| 1. | <code>&mas[i]</code> | А. <code>mas+i;</code>
// ТАК |
| 2. | <code>mas[i]</code> | Б. ДЕЛАТЬ НЕЛЬЗЯ |

3. mas++;

В. *
 (mas+i);
Г. *mas

№ 3 Прочитайте текст и установите последовательность

Расположите фрагменты программы в таком порядке, чтобы в результате на экран была выведена строка "Hello, World!" в верхнем регистре ("HELLO, WORLD!").

1. printf("%s", str);
2. char str[20] = "Hello, World!"
3. while(str[i]) {
4. i++;
5. str[i] = toupper(str[i]);
6. int i = 0;
7. }

№ 4 Прочитайте текст и установите последовательность

Расположите фрагменты программы в таком порядке, чтобы программа корректно вычисляла и выводила сумму двух чисел с использованием функции.

1. int num1 = 5, num2 = 3;
2. return a + b;
3. printf("Сумма: %d", result);
4. int main() {
5. int sum(int a, int b) {
6. return 0; }
7. }
8. int result = sum(num1, num2);

№ 5 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какая функция стандартной библиотеки C корректно определяет длину строки, учитывая завершающий ноль-символ \0?

1. sizeof(str)
2. strlen(str)
3. strcount(str)
4. strlenlength(str)

№ 6 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

В программе объявлены следующие строки:

char str1[20] = "Hello";

char str2[20] = "World";

Какие из приведённых операций корректно объединяют строки str1 и str2 в str1 (результат: "HelloWorld") без возникновения ошибок?

1. strcat(str1, str2);
2. str1 = str1 + str2;

```

3. int i = strlen(str1);

   int j = 0;

   while(str2[j] != '\0') {

       str1[i] = str2[j];

       i++;

       j++;

   }

   str1[i] = '\0';

```

4. `sprintf(str1, "%s%s", str1, str2);`

№ 7 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих утверждений о функциях в языке С являются верными?

Варианты ответов:

1. Функция может возвращать несколько значений одновременно
2. Рекурсивные функции не могут вызывать другие функции
3. Указатели могут использоваться для передачи данных по ссылке
4. Статические переменные внутри функции сохраняют свое значение между вызовами
5. Все функции должны быть объявлены до их использования
6. Функция `main()` должна возвращать значение типа `int`

№ 8 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие из следующих утверждений о структурных типах данных в языке С++ являются верными?

1. Структуры (`struct`) и классы (`class`) в С++ идентичны, за исключением уровня доступа по умолчанию
2. Размер структуры всегда точно равен сумме размеров всех её членов
3. Для доступа к элементам структуры можно использовать как оператор точки (`.`), так и оператор стрелки (`->`)
4. Структуры нельзя передавать в функции по значению
5. Объявление структуры не занимает память до создания объекта этого типа
6. Структуры могут содержать методы (функции-члены)
7. В С++ структуры не могут наследоваться от других структур

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какой режим открытия файла в языке С используется, если нужно:

- создать новый файл (если его нет),
- перезаписать содержимое (если файл существует),
- разрешить только запись?

Варианты ответов:

1. "r"
2. "w"
3. "a"
4. "r+"

№ 10 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какое ключевое слово используется в языке С для возврата значения из функции?

1. break
2. exit
3. return
4. continue

№ 11 Прочитайте текст и запишите развернутый обоснованный ответ

Что такое структуры (struct) в C++? Как их объявлять?

№ 12 Прочитайте текст и запишите развернутый обоснованный ответ

Что такое файловый ввод-вывод в C++? Какие классы и операции используются для работы с файлами? Покажите на примере, как записать данные в файл и прочитать их обратно

ОПК-9 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

№ 1 Прочитайте текст и установите последовательность

Установите правильную последовательность действий при работе со структурными типами данных:

1. Объявление структурного типа с помощью ключевого слова struct.
2. Определение битовых полей для экономии памяти.
3. Инициализация объекта структуры.
4. Объявление указателя на структурный тип.
5. Динамическое выделение памяти для структурного объекта.
6. Доступ к полям структуры через указатель с использованием ->.
7. Использование union для хранения разных типов данных в одной области памяти.

№ 2 Прочитайте текст и установите последовательность

Расположите шаги в правильном порядке:

1. Создание переменной структурного типа.
2. Присваивание адреса структурной переменной указателю.
3. Освобождение памяти, если она выделялась динамически.
4. Объявление структуры с полями разных типов.
5. Объявление указателя на структурный тип.
6. Изменение значения поля структуры через указатель.
7. Инициализация полей структуры с использованием точечной нотации.

№ 3 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

В каком случае функция возвращает целое число и ничего не принимает в качестве входных параметров?

1. int func(void);
2. void func(int);
3. void* func(char);
4. void func(void);

№ 4 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Выберите верные утверждения:

1. При добавлении элемента в начало односвязного списка новый узел становится головой списка, а его указатель связи (next) устанавливается на старую голову.
2. Удаление элемента из середины односвязного списка требует только освобождения памяти, так как связи между узлами корректируются автоматически.
3. Для удаления последнего элемента односвязного списка необходимо пройти весь список до предпоследнего узла и изменить его указатель связи (next) на NULL.
4. При добавлении элемента в конец односвязного списка не требуется проход по всем узлам, если хранится указатель на хвост.

№ 5 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Выберите верные утверждения:

1. Сортировка односвязного списка методом пузырька менее эффективна, чем сортировка вставками, из-за частых перестановок.
2. Для поиска элемента в двусвязном списке можно использовать как линейный поиск с головы, так и с хвоста.
3. Бинарный поиск применим к односвязным спискам, так как они допускают быстрый доступ к середине последовательности.
4. Удаление элемента из отсортированного списка не нарушает порядок сортировки оставшихся элементов.

№ 6 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Выберите верные утверждения:

1. В двусвязном списке каждый узел содержит два указателя: на следующий (next) и на предыдущий (prev) элемент.
2. Удаление элемента из двусвязного списка всегда требует только освобождения памяти, так как связи корректируются автоматически.
3. Добавление элемента в конец двусвязного списка выполняется за $O(1)$, если хранится указатель на хвост (tail).
4. Поиск элемента в двусвязном списке всегда быстрее, чем в односвязном, так как можно двигаться в обоих направлениях.

№ 7 Прочитайте текст и установите соответствие

Соотнесите функции работы с бинарными файлами и их описания:

1	"wb"	А	Читает блок данных из бинарного файла
2	fwrite()	Б	Режим записи бинарного файла (создание/ перезапись)
3	fread()	В	Записывает блок данных в бинарный файл
4	fseek()	Г	Перемещает указатель позиции в файле
		Д	Возвращает текущую позицию указателя в файле

№ 8 Прочитайте текст и установите соответствие

Соотнесите функции работы с текстовыми файлами и их описания:

1	FILE *fopen()	А	Закрывает файл и освобождает указатель
2	fprintf()	Б	Открывает файл в указанном режиме ("r", "w")

3 fclose()

В Читает строку из
файла

4 "a"

Записывает
Г форматированные
данные в файл
Режим
Д дописывания в
конец файла

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Имеется прототип функции:

```
void func (double, int *);
```

В вызывающей функции объявлены переменные

```
double a, *p;
```

```
int b;
```

Какой из предложенных вариантов вызова объявленной функции является синтаксически верным?

1. func (a, &a);
2. func (*p, &p);
3. func (a, b+p);
4. func (a, &b);

№ 10 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Укажите правильное объявление указателя на переменную с плавающей точкой в C:

1. float &x;
2. float *x;
3. int x;
4. ptr x;

№ 11 Прочитайте текст и запишите развернутый обоснованный ответ

Объясните разницу между формальными и фактическими параметрами функции. Приведите пример.

№ 12 Прочитайте текст и запишите развернутый обоснованный ответ

Опишите, как объявить и использовать указатель на функцию для вычисления интеграла. Почему это эффективнее прямого вызова функции?