



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

«УТВЕРЖДАЮ»

и.о. проректора

А.В. Троицкий

«__»_____2023г.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ**

**КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев 2023

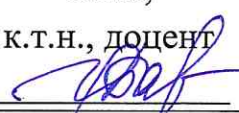
Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе основной профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор: к.т.н. Исаева Г.Н. Рабочая программа дисциплины: Языки программирования высокого уровня– Королев МО: «Технологический университет», 2023.

Рецензент: к.т.н. доцент Логачёва Н.В..

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки бакалавров 09.03.04 Программная инженерия, учебного плана, утвержденного Ученым советом Технологического университета, Протокол № 9 от 11 апреля 2023 года.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Баранова О.М., к.т.н., доцент 			
Год утверждения (переутверждения)	2023	2024	2025	2026
Номер и дата протокола заседания кафедры	№ 7 от 03.04.2023			

Рабочая программа согласована:

Руководитель ОПОП  О.М. Баранова, к.т.н.

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2023	2024	2025	2026
Номер и дата протокола заседания УМС	№ 5 от 11.04.2023 г.			

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП

Целью изучения дисциплины является:

- развитие у студентов личностных качеств;
- формирование компетенций в соответствии с действующим образовательным стандартом по направлению подготовки 09.03.04 «Программная инженерия».

Дисциплина направлена на формирование следующих компетенций:

- (ОПК-2) Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;
- (ОПК-3) Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- (ОПК-6) Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
- (ОПК-7) Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой.

Основными задачами дисциплины являются:

— подготовка обучающихся к *научно-исследовательской деятельности*: проведение научно-исследовательских мероприятий в рамках анализа качества программного продукта, определение возможных способов повышения качества, разработка комплекса мероприятий по повышению качества программного продукта;

— подготовка обучающихся к *проектной деятельности*: определение требований к программному продукту, разработка проекта программного продукта с использованием известных способов и методов проектирования, разработка проектной документации;

— подготовка обучающихся к *производственно-технологической деятельности*: реализация программного продукта на языке программирования с использованием известных технологий и методов программирования, тестирование, интеграция программных продуктов, повышение качества программного продукта, сопровождение программного продукта.

Показатель освоения компетенции отражают следующие индикаторы:

Трудовые действия:

Применяет современные информационные технологии и программные средства, в том числе, отечественного производства при решении задач профессиональной деятельности;

Применяет принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

Анализирует информационно-коммуникационных технологий и с учетом основных требований информационной безопасности для подготовки обзоров, аннотаций, составления рефератов, научных докладов;

Применяет основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий;

Применяет в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;

Необходимые умения:

Анализирует современные информационные технологии и программные средства, в том числе, отечественного производства при решении задач профессиональной деятельности;

Использует современные информационные технологии и программные средства, в том числе, отечественного производства, при решении задач профессиональной деятельности;

Использует знания об основах информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности для решения стандартных задач профессиональной деятельности;

Анализирует современные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды для разработки информационных систем и технологий;

Использует современные языки программирования, программные среды разработки информационных систем и технологий для отладки и тестирования прототипов программно-технических комплексов задач;

Использует основные концепции, принципы, теории и факты, связанные с информатикой для решения практических задач;

Необходимые знания:

Знает принципы работы современных информационных технологий и программных средств, в том числе отечественного производства;

Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий;

Знает основные концепции, принципы, теории и факты, связанные с информатикой.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина Б1.О.17.01 «Языки программирования высокого уровня» относится к обязательной части основной профессиональной образовательной программы подготовки по направлению подготовки бакалавриата 09.03.04 «Программная инженерия».

Дисциплина реализуется кафедрой прикладного программного обеспечения.

Изучение данной дисциплины базируется на знаниях и умениях, полученных в рамках ранее изученных дисциплин «Информатика», «Основы алгоритмизации и программирования», «Информационные технологии в профессиональной деятельности», «Адаптированные информационные технологии» и компетенциях УК-2, УК-6, ОПК-2, ОПК-3, ОПК-6, ОПК-7, ОПК-8..

3. Объем дисциплины (модуля) и виды учебной работы

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

. Таблица 1

Виды занятий	Всего часов	Семестр третий	Семестр четвертый	Семестр ...	Семестр ...
Общая трудоемкость	216	108	108		
Аудиторные занятия	96	48	48		
Лекции (Л)	32	16	16		
Практические занятия (ПЗ)	64	32	32		
Лабораторные работы (ЛР)	-	-	-		
Практическая подготовка	-	-	-		
Самостоятельная работа	120	60	60		
Курсовые, расчетно-графические работы	-	-	-		
Контрольная работа	+	+	+		
Текущий контроль знаний (7 - 8, 15 - 16 недели)	Тест	Тест	Тест		
Вид итогового контроля	Зачет/Экзамен	Зачёт	Экзамен		

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час	Занятия в интерактивной форме, час	Код компетенций
Третий семестр				
Тема 1. Введение в курс «Языки программирования»	2	2	-	ОПК-7
Тема 2. История создания ЯП ВУ. Состав ЯП	4	8	2	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Тема 3. Структурное и модульное программирование	8	10	2	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Тема 4. Основные принципы программирования на языках высокого уровня	2	12	4	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Четвёртый семестр				
Тема 5. Языки и системы программирования ООП - парадигмы	4	4	2	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Тема 6. Технология объектно-ориентированного программирования	2	6	2	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Тема 7 Стандартные библиотеки и их подпрограммы в C++.	6	16	2	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Тема 8. Языки и системы функционального и логического программирования	4	6	2	ОПК-2, ОПК-3, ОПК-6, ОПК-7
Итого:	32	64	16	

4.2. Содержание тем дисциплины

Тема 1. Введение в курс «Языки программирования».

Основные понятия. Язык программирования (ЯП). Программное обеспечение (ПО). Система программирования.

Тема 2. История создания ЯП ВУ. Состав ЯП

Многообразие современных языков программирования. Концепция языков программирования. Классификация языков программирования по степени детализации предписаний. История языков программирования высокого уровня (ЯП ВУ). Спецификация программ и стандартизация ЯП. Реализация языка программирования. Си - подобный синтаксис. Программирование вычислений.

Тема 3. Структурное и модульное программирование

Типы данных. Структуры данных. Представление основных управляющих структур программирования. Процедуры и функции. Модули. Модули ввода-вывода. Библиотечные функции.

Тема 4. Основные принципы программирования на языках высокого уровня.

Основные подходы к построению структурного кода. Этапы решения задач на ЭВМ: тестирование и отладка программ. Сопровождение программы
Критерии качества программы.

Тема 5. Языки и системы программирования ООП – парадигмы

Основные принципы объектно-ориентированного программирования. Вычислительная модель. Языки ООП и современные среды. Классы и объекты. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.

Тема 6. Технология объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Вычислительная модель. Языки ООП и современные среды. Конструкторы и деструкторы: конструкторы по умолчанию, с аргументами и конструктор копирования, их назначение

Тема 7 Стандартные библиотеки и их подпрограммы в С++ Перегрузка операций в классе. Агрегированные классы. Обработка исключительных ситуаций. Основные контейнеры.

Тема 8.. Языки и системы функционального и логического программирования. Вычислительная модель функционального программирования. Вычислительная модель логического программирования. Мультипарадигмальность. ЯП ВУ данных парадигм.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

Методические указания для самостоятельной работы обучающихся по освоению дисциплины представлены в Приложении 2.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Структура фонда оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведена в Приложении 1.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. С++ как второй язык в обучении приемам и технологиям программирования: учебное пособие / Я.М. Русанова. - Ростов-на-Дону: Издательство ЮФУ, 2010. - 200 с. ISBN 978-5-9275-0749-8
<http://znanium.com/catalog.php?item=bookinfo&book=550811>

2. Бедердинова О. И. Программирование на языках высокого уровня: учеб. пособие / О.И. Бедердинова, Т.А. Минеева, Ю.А. Водовозова. — Москва: ИНФРА-М, 2019. — 159 с. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1044396>

3. Структуры и алгоритмы обработки данных: Учебное пособие / В.Д. Колдаев. - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014. - 296 с.: 60x90 1/16. - (Высшее образование: Бакалавриат). (переплет) ISBN 978-5-369-01264-2, 500 экз. <http://www.znanium.com/catalog.php>

4. Белов В.В., Чистякова В.И. Алгоритмы и структуры данных: Учебник М.: КУРС, НИЦ ИНФРА-М, 2016. - 240 с. ISBN 978-5-906818-25-6. / ЭБС «Знаниум».

Дополнительная литература:

1. Колдаев В.Д. Численные методы и программирование: учеб. пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 336 с.: ил. — (Профессиональное образование).

<http://znanium.com/catalog.php?item=bookinfo&book=672965>

2. Программирование графики на C++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 517 с. + Доп. материалы [Электронный ресурс; Режим доступа <http://www.znanium.com>]. — (Высшее образование). — www.dx.doi.org/10.12737/23113.

<http://znanium.com/catalog.php?item=bookinfo&book=562914>

3. Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке C++: Учебное пособие. - 1. - Москва; Москва: Издательский Дом "ФОРУМ": ООО "Научно-издательский центр ИНФРА-М", 2018. - 512 с. URL: <http://znanium.com/go.php?id=918098>

4. Языки программирования: учеб. пособие / О.Л. Голицына, Т.Л.Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — М.: ФОРУМ: ИНФРА-М, 2018. — 399 с. URL: <http://znanium.com/bookread2.php?book=973007>

5. Галимов, Р. Р. Программирование в среде Visual C++ с использованием библиотеки MFC: метод. указания к лаб. работам / А. И. Сарайкин, Оренбургский гос. унт, Р. Р. Галимов. — Оренбург: ОГУ, 2014 <http://rucont.ru/efd/271378?cldren=0>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

<http://www.znanium.com/> – электронно-библиотечная система Znanium;
<http://www.e.lanbook.com/> – ЭБС Издательства «ЛАНЬ»;

<http://www.rucont.ru/> – электронно-библиотечная система;
<http://www.biblioclub.ru/> – университетская библиотека онлайн.

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в Приложении 2.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения:

Прикладные программы CODE::Blocks, Visual Studio, ППП ONLYOFFICE.

Информационные справочные системы:

1. Электронные ресурсы образовательной среды Университета.
2. Рабочая программа и методическое обеспечение по дисциплине «Языки программирования высокого уровня»

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия:

– аудитория, оснащенная презентационной техникой (проектор, экран), интерактивной доской Smart Board.

Практические занятия:

– учебный класс, оснащенный вычислительной техникой (ПК), современные лицензионные программно-технические средства: операционная система не ниже Windows 10; офисные программы и среды программирования, поддерживающие ЯП C/C++.

– рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;

– рабочие места студентов, оснащенные компьютерами с доступом в Интернет.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХ-
НОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕ-
ЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ
«ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ»**

(Приложение 1 к рабочей программе)

Направление подготовки: 09.03.04 Программная инженерия

Профиль: Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции, обучающийся приобретает:		
				Трудовые действия	Необходимые умения	Необходимые знания
1.	ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;	Тема 2 Тема 3 Тема 4 Тема 5 Тема 6 Тема 7 Тема 8	Применяет современные информационные технологии и программные средства, в том числе, отечественного производства при решении задач профессиональной деятельности.	Анализирует современные информационные технологии и программные средства, в том числе, отечественного производства при решении задач профессиональной деятельности; Использует современные информационные технологии и программные средства, в том числе, отечественного производства, при решении задач профессиональной деятельности;	Знает принципы работы современных информационных технологий и программных средств, в том числе отечественного производства
2.	ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Тема 2 Тема 3 Тема 4 Тема 5 Тема 6 Тема 7 Тема 8	Применяет принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Использует знания об основах информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности для решения стандартных задач профессиональной деятельности;	Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

				Анализирует информационно-коммуникационных технологий и с учетом основных требований информационной безопасности для подготовки обзоров, аннотаций, составления рефератов, научных докладов		
3.	ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;	Тема 2 Тема 3 Тема 4 Тема 5 Тема 6 Тема 7 Тема 8	Применяет основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий;	Анализирует современные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды для разработки информационных систем и технологий Использует современные языки программирования, программные среды разработки информационных систем и технологий для отладки и тестирования прототипов программно-технических комплексов задач;	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
4.	ОПК-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	Тема 1 Тема 2 Тема 3 Тема 4 Тема 5 Тема 6 Тема 7 Тема 8	Применяет в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	Использует основные концепции, принципы, теории и факты, связанные с информатикой для решения практических задач	Знает основные концепции, принципы, теории и факты, связанные с информатикой

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Инструменты, оценивающие сформированность компетенции	Показатель оценивания компетенции	Критерии оценки
ОПК-3, ОПК-7	Доклад в форме презентации	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (компетенция не сформирована) – 2 и менее баллов</p>	<p>Проводится устно с использованием мультимедийных систем, а также с использованием технических средств</p> <p>Время, отведенное на процедуру – 10 - 15 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <ol style="list-style-type: none"> 1.Соответствие представленной презентации заявленной тематике (1 балл). 2.Качество источников и их количество при подготовке доклада и разработке презентации (1 балл). 3.Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4.Качество самой представленной презентации (1 балл). 5.Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл). <p>Максимальная сумма баллов - 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.</p>
ОПК-2, ОПК-3, ОПК-6	Расчетно-графические работы	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (компетенция не сформирована) – 2 и менее баллов</p>	<p>Проводится устно с использованием мультимедийных систем, а также с использованием технических средств</p> <p>Время, отведенное на процедуру – 10 - 15 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <ol style="list-style-type: none"> 1.Соответствие представленной презентации заявленной тематике (1 балл). 2.Качество источников и их количество при подготовке доклада и разработке презентации (1 балл). 3.Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4.Качество самой представленной презентации (1 балл). 5.Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл). <p>Максимальная сумма баллов - 5 баллов.</p>

			Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.
ОПК-2, ОПК-3, ОПК-6 ОПК-7	Контрольная работа	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (компетенция не сформирована) – 2 и менее баллов</p>	<p>1. Проводится в форме письменной работы</p> <p>2. Время, отведенное на процедуру – семестр.</p> <p>Неявка на защиту контрольной работы – 0.</p> <p>Критерии оценки:</p> <p>1. Соответствие содержания контрольной работы заявленной тематике (1 балл).</p> <p>2. Качество источников и их количество при подготовке работы (1 балл).</p> <p>3. Владение информацией и способность отвечать на вопросы аудитории (1 балл).</p> <p>4. Качество самой представленной работы (1 балл).</p> <p>5. Использование специализированного программного обеспечения (1 балл).</p> <p>6. Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл).</p> <p>Максимальная сумма баллов - 6 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.</p>

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1. Примерная тематика докладов в форме презентации (3 семестр)

1. Язык программирования как знаковая система.
2. Современная концепция мультипарадигмальности ЯПВУ.
3. Язык программирования Фортран.
4. Язык программирования Алгол
5. Формализация задач объективного мира.

6. Конвертор – тип транслятора.
7. Язык программирования Perl
8. Язык программирования РЕФАЛ
9. Язык программирования Python
10. Язык программирования Ruby
11. Язык программирования Scala
12. Язык программирования Оберон
13. Язык программирования Симула
14. Язык программирования Ladder Diagram (LD)
15. Широта использования современных языков и систем программирования.
16. Основные парадигмы и концепции программирования в свете ЯПВУ.
17. Язык программирования АДА
18. Простые рекурсивные функции.
19. Библиотеки работы с графикой в популярных средах программирования.
20. Язык программирования C#.
21. Язык программирования Java.
22. Язык программирования Visual Basic
23. Язык программирования Haskell.
24. Среды программирования, поддерживающие функциональную парадигму.
25. Среды программирования, поддерживающие логическую парадигму.

3.2. Примерные задания расчетно-графических работ (4 семестр)

Во всех вариантах должны быть использованы функции для логически законченных фрагментов программного кода; должен присутствовать удобный интерфейс для выбора операций с двусвязным списком..	
№ Варианта	Словесное описание задачи
1.	Построить линейный список из входной последовательности чисел. Удалить из него все числа, кратные 5.

2.	Построить линейный список из входной последовательности чисел. Удалить из него все числа, кратные 3.
3.	Построить линейный список из входной последовательности чисел. Поменять местами первое и максимальное число, последнее и минимальное.
4.	Построить линейный список из входной последовательности чисел. Поменять местами первое и второе число, третье и четвертое и т. д.
5.	Построить линейный список из входной последовательности чисел. Поменять местами первое и последнее число
6.	Построить линейный список из входной последовательности чисел. Удалить из него все числа, кроме простых.
7.	Построить линейный список из входной последовательности чисел. Удалить из него все простые числа.
8.	Построить линейный список из входной последовательности чисел. Перестроить его так, чтобы сначала стояли все нечетные числа, затем все четные.
9.	Построить линейный список из входной последовательности чисел. Перестроить его так, чтобы сначала стояли все четные числа, затем все нечетные
10.	Построить линейный список из входной последовательности чисел. Удалить из него все нечетные числа.
11.	Построить линейный список из входной последовательности чисел. Удалить из него все четные числа.
12.	Построить линейный список из входной последовательности чисел. Перестроить его так, чтобы сначала стояли все отрицательные числа, затем все положительные.
13.	Построить линейный список из входной последовательности чисел. Перестроить его так, чтобы сначала стояли все положительные числа, затем все отрицательные.
14.	Построить линейный список из входной последовательности чисел. Удалить из него все положительные числа
15.	Построить линейный список из входной последовательности чисел. Удалить из него все отрицательные числа.
16.	Построить линейный список из фамилий. Оставить в нем только фамилии, состоящие из 5 букв
17.	Построить линейный список из фамилий. Оставить в нем только фамилии, начинающиеся на букву «А», остальные удалить.
18.	Построить линейный список из фамилий. Удалить из него фамилии, начинающиеся на букву «А».

19.	Построить линейный список из фамилий. Удалить из него самую длинную фамилию.
20.	Построить линейный список из фамилий. Удалить из него самую короткую фамилию.
21.	Построить линейный список из фамилий. Вставить в него введенную фамилию, сохранив алфавитный порядок.
22.	Построить линейный список из фамилий. Оставить в нем только фамилии, заканчивающиеся на сочетание «ов», остальные удалить
23.	Построить линейный список из фамилий. Оставить в нем только фамилии, заканчивающиеся на букву «а», остальные удалить.
24.	Построить линейный список из входной последовательности чисел. Поменять местами первое положительное число и последний ноль
25.	Построить линейный список из 20 чисел, значения которых лежат в пределах от 0 до 1000. Удалить из него все двузначные числа.

3.3. Примерная тематика заданий контрольной работы

Решение практических задач методом ООП в интегрированной среде разработки приложений MS Visual Studio на ЯП ВУ C++	
Цель работы – получить практические навыки работы с абстрактными, базовыми и производными классами, методами классов. Правильно создавать и использовать объекты.	
Во всех вариантах стараться использовать меню, оформленное как класс.	
№ Варианта	Словесное описание задачи
1.	Создать класс CFile, инкапсулирующий в себе такие функции работы с файлами, как Open, Close, Seek, Read, Write, GetPosition и GetLength. На базе этого класса создать производный класс CMyDataFile — файл, содержащий в себе данные некоторого определенного типа MyData, а также заголовок, облегчающий доступ к этому файлу. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.
2.	Создать класс CPoint — точка. На его основе создать классы CcoloredPoint и CLine. На основе класса CLine создать класс CColoredLine и класс CPolyLine — многоугольник. Все классы должны иметь методы для установки и получения значений всех координат, а также перемещения и поворота фигур.

	<p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
3.	<p>Создать абстрактный класс <code>CVehicle</code>. На его основе реализовать классы <code>CPlane</code>, <code>CSaг</code> и <code>CShip</code>. Классы должны иметь возможность задавать и получать координаты, параметры средств передвижения (цена, скорость, год выпуска). Для самолета должна быть определена высота, для самолета и корабля — количество пассажиров.</p> <p>Для корабля — порт приписки.</p> <p>Написать программу, создающую список объектов этих классов. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
4.	<p>Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии или номеру телефона), добавления и удаления записей.</p> <p>Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.</p>
5.	<p>Описать базовый класс <code>СТРОКА</code>.</p> <p>Обязательные поля класса:</p> <ul style="list-style-type: none"> • значение типа <code>char</code> — хранит строку; • значение типа <code>int</code> — хранит длину строки в байтах. <p>Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор, принимающий в качестве параметра символьную строку (заканчивается пулевым байтом); • конструктор, принимающий в качестве параметра символ; • конструктор копирования; • получение длины строки; • очистка строки (сделать строку пустой); <p>Описать производный от <code>СТРОКА</code> класс <code>ПРЕДЛОЖЕНИЕ</code>.</p> <p>Обязательные методы:</p> <ul style="list-style-type: none"> • перевод всех символов строки(предложения) в верхний регистр; • перевод всех символов строки (предложения) в нижний регистр; • поиск первого вхождения символа в строку; <p>Определить следующие операции:</p> <ul style="list-style-type: none"> • присваивание (=); • сложение (+) — операция конкатенации строк – получение предложения; • вычитание (-) — из строки (первый операнд) удаляются все символы, входящие в строку — второй операнд, при этом может получиться пустая строка(предложение). <p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
6.	<p>Создать класс <code>CPoint</code> — <code>ТОЧКА</code>. На его основе создать производные классы <code>CircleLine</code> и <code>DirectLine</code>. На основе класса <code>CircleLine</code> предусмотреть возможность создания объектов разного диаметра и положения на плоскости;</p> <p>На основе класса <code>DirectLine</code> — квадраты, прямоугольники, многоугольники.</p> <p>Все классы должны иметь методы для установки и получения значений всех координат, а также изменения цвета и получения текущего цвета.</p>

	<p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
7.	<p>Описать базовый класс СТРОКА, Обязательные поля класса:</p> <ul style="list-style-type: none"> • значение типа char — хранит символы строки; • значение типа int — хранит длину строки в байтах. <p>Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор, принимающий в качестве параметра исходную символьную строку (заканчивается нулевым байтом); • конструктор, принимающий в качестве параметра символ; • конструктор копирования; • получение длины строки; • очистка строки (сделать строку пустой); <p>Описать производный от СТРОКА класс ДЕСЯТИЧНАЯ_СТРОКА. Строки данного класса могут содержать только символы десятичных цифр и символы - и +, задающие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, ДЕСЯТИЧНАЯ_СТРОКА принимает нулевое значение. (Например, -345h – после отработки идентификации числовой строки – выдается 0) Содержимое строк рассматривается как десятичное число. Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор без параметров; • конструктор, принимающий в качестве параметра символьную -строку (заканчивается нулевым байтом); • конструктор копирования; • метод, определяющий, можно ли представить данное число в формате int; <p>Определить следующие операции:</p> <ul style="list-style-type: none"> • присваивание (=); • вычитание (-) — арифметическая разность строк; • операция (>) — проверка на больше (по значению); • операция (<) — проверка на меньше (по значению). <p>Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
8.	<p>Описать базовый класс СТРОКА, Обязательные поля класса:</p> <ul style="list-style-type: none"> • значение типа char — хранит символы строки; • значение типа int — хранит длину строки в байтах. <p>Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор, принимающий в качестве параметра исходную символьную строку (заканчивается нулевым байтом); • конструктор, принимающий в качестве параметра символ; • конструктор копирования; • получение длины строки; • очистка строки (сделать строку пустой); <p>Описать производный от СТРОКА класс КОМПЛЕКСНОЕ_ЧИСЛО. Строки данного класса состоят из двух полей, разделенных символом i.</p>

	<p>Первое поле задает значение реальной части числа, а второе — мнимой. Каждое из полей может содержать только символы десятичных цифр и символы - и +, задающие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, КОМПЛЕКСНОЕ_ЧИСЛО принимает нулевое значение. Примеры строк: 33112, -7i100, +5i-21.</p> <p>Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор, принимающий в качестве параметра символьную строку (заканчивается нулевым байтом); • конструктор копирования; <p>Обеспечить следующие операции:</p> <ul style="list-style-type: none"> • присваивание (=); • операция (==) — проверка на равенство; • умножение (*) — умножение чисел. <p>Программа должна содержать меню, позволяющее осуществить проверку всех методов классов</p>
9.	<p>Разработать программу, имитирующую работу функционального калькулятора (который позволяет выбрать в диалоге какую-либо из известных ему функций, затем предлагает ввести значение аргумента и, возможно, коэффициентов и после ввода выдает соответствующее значение функции).</p> <p>В первой версии калькулятора «база знаний» содержит две функции:</p> <ul style="list-style-type: none"> • экспоненту $y = e^x$; • линейную функцию $y = ax + b$. <p>Далее нужно создать класс ПРОИЗВОДНЫЙ КАЛЬКУЛЯТОР и добавить в него функции, использующие в качестве базовых две функции родительского класса.</p> <p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
10.	<p>Разработать программу, имитирующую работу функционального КАЛЬКУЛЯТОРА, который позволяет выбрать с помощью меню какую-либо из известных ему функций, затем предлагает ввести значение аргумента и, возможно, коэффициентов и после ввода выдает соответствующее значение функции.</p> <p>В первой версии калькулятора «база знаний» содержит три функции:</p> <ul style="list-style-type: none"> • экспоненту $y = e^x$; • линейную функцию $y = ax + b$. • параболу $y = ax^2 + bx + c$ <p>Далее нужно создать класс ПРОИЗВОДНЫЙ КАЛЬКУЛЯТОР и добавить в него функции, использующие в качестве базовых функции родительского класса.</p> <p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов</p>
11.	<p>Разработать программу, имитирующую работу С КАЛЬКУЛЯТОРОМ.</p> <p>Предусмотреть методы:</p> <ul style="list-style-type: none"> - сложения, вычитания, возведения в степень - выдачи сообщений о неверных действиях на основе методов обработки исключений в Java/

	<p>Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.</p>
12.	<p>Описать базовый класс СТРОКА, Обязательные поля класса:</p> <ul style="list-style-type: none"> • указатель на char — хранит символы строки; <p>значение типа int — хранит длину строки в байтах. Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор, принимающий в качестве параметра исходную символьную строку (заканчивается нулевым байтом); • конструктор, принимающий в качестве параметра символ; • конструктор копирования; • получение длины строки; • очистка строки (сделать строку пустой); <p>Описать производный от СТРОКА класс КОМПЛЕКСНОЕ__ЧИСЛО. Строки данного класса состоят из двух полей, разделенных символом i. Первое поле задает значение реальной части числа, а второе — мнимой. Каждое из полей может содержать только символы десятичных цифр и символы - и +, задающие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, КОМПЛЕКСНОЕ_ЧИСЛО принимает нулевое значение. Примеры строк: 33112, -7i100, +5i-21. Обязательные методы:</p> <ul style="list-style-type: none"> • конструктор, принимающий в качестве параметра символьную строку (заканчивается нулевым байтом); • конструктор копирования; <p>Обеспечить следующие операции:</p> <ul style="list-style-type: none"> • присваивание (=); • операция (==) — проверка на равенство; • сложение (+) — сложение чисел. • вычитание (-) — вычитание чисел. <p>Программа должна содержать меню, позволяющее осуществить проверку всех методов классов</p>
13.	<p>Создать класс File, инкапсулирующий в себе такие функции работы с файлами, как Open, Close, Seek, Read, Write, GetPosition и GetLength. На базе этого класса создать производный класс MyDataFile — файл, содержащий в себе записи определенного типа о студенческой группе MyData, а также заголовок, облегчающий доступ к этому файлу. Написать программу, демонстрирующую работу с этими классами. Программа должна: содержать меню, позволяющее осуществить проверку всех методов классов.</p>
14.	<p>Создать абстрактный класс ЖИВОТНЫЕ. На его основе реализовать классы СУХОПУТНЫЕ, ПЛАВАЮЩИЕ и ЛЕТАЮЩИЕ. Классы должны иметь возможность задавать и получать характеристики животных, согласно среде их обитания; а также параметры передвижения (скорость плавания/ летания/ прыгания и т.д.). Написать программу, создающую список объектов этих классов. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>

15.	<p>Создать класс CFile, инкапсулирующий в себе такие функции работы с файлами, как Open, Close, Seek, Read, Write, GetPosition и GetLength. На базе этого класса создать производный класс CMyDataFile — файл, содержащий в себе данные некоторого определенного типа MyData, а также заголовок, облегчающий доступ к этому файлу.</p> <p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
16.	<p>Построить описание класса АДРЕС, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.</p> <p>Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.</p>
17.	<p>Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.</p> <p>Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.</p>
18.	<p>Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям.</p> <p>Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.</p>
19.	<p>Разработать программу, имитирующую работу функционального калькулятора (который позволяет выбрать в диалоге какую-либо из известных ему функций, затем предлагает ввести значение аргумента и, возможно, коэффициентов и после ввода выдает соответствующее значение функции).</p> <p>В первой версии калькулятора «база знаний» содержит две функции:</p> <ul style="list-style-type: none"> • экспоненту $y = e^x$; • параболу $y = ax^2 + bx + c$ <p>Далее нужно создать класс ПРОИЗВОДНЫЙ КАЛЬКУЛЯТОР и добавить в него функции, использующие в качестве базовых две функции родительского класса.</p> <p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
20.	<p>Создать класс CPoint — ТОЧКА. На его основе создать производные классы CircleLine и DirectLine. На основе класса CircleLine предусмотреть возможность создания объектов разного диаметра и положения на плоскости;</p> <p>На основе класса DirectLine — многоугольники.</p> <p>Все классы должны иметь методы для установки и получения значений всех координат, а также изменения положения фигур на плоскости:</p>

	<p>- сдвига; - поворота.</p> <p>Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.</p>
21.	<p>Создать абстрактный класс РАСТЕНИЯ. На его основе реализовать классы хвойные, лиственные и смешанные. Классы должны иметь возможность задавать и получать характеристики растений, согласно среде их обитания; а также параметры существования (скорость роста/ скорость плод созревания, цветения и т.д.). Написать программу, создающую список объектов этих классов. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов</p>

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формой контроля знаний по дисциплине «Языки программирования высокого уровня» являются две текущие аттестации в форме тестов и итоговые аттестации в формах зачёта в третьем семестре и экзамена в четвёртом семестре.

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
7-8	тестирование	ОПК-2, ОПК-3 ОПК-6 ОПК-7	25 вопросов	Компьютерное тестирование; время, отведенное на процедуру -30 минут,	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - Удовлетворительно - от 51% правильных ответов. Хорошо - от 75%. Отлично – от 85%.
15-16	тестирование	ОПК-2, ОПК-3 ОПК-6 ОПК-7	25 вопросов	Компьютерное тестирование; время, отведенное на процедуру -30 минут,	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - Удовлетворительно - от 51% правильных ответов. Хорошо - от 75%. Отлично – от 85%.

В соответствии с КУГ	Зачёт	ОПК-2, ОПК-3 ОПК-6 ОПК-7	2 вопроса	Зачёт проводится в письменной форме, путем ответа на вопросы. Время, отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения зачёта	Критерии оценки: «Зачтено»: <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета. «Не зачтено»: <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; • не отвечает на вопросы.
В соответствии с КУГ	Экзамен	ОПК-2, ОПК-3 ОПК-6 ОПК-7	3 вопроса	Экзамен проводится в письменной форме, путем ответа на вопросы. Время, отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения экзамена	Критерии оценки: «Отлично»: <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета. «Хорошо»: <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и

						<p>применять полученные знания на практике;</p> <ul style="list-style-type: none"> • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета <p>• неправильно решено практическое задание</p> <p>«Удовлетворительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; <p>«Неудовлетворительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; • не отвечает на вопросы.
--	--	--	--	--	--	--

4.1. Типовые вопросы, выносимые на тестирование

Тесты используются в режиме промежуточного контроля. По форме заданий выбраны закрытые тесты (с выборочным ответом). Каждому вопросу соответствует один вариант ответа.

1. Основное требование, предъявляемое к программе:

надежность
сложность
простота
компактность

2. Какое программирование называют «стиль спагетти»?

программирование с бесконечными переходами
программирование на языке низкого уровня
программирование без «goto»
программирование на языке высокого уровня

3. Создание любой программы начинается...

с постановки задачи
разработки внутренних структур данных
со структурного программирования
с проектирования

4. Создателем языка Паскаль является ...

Никлаус Вирт
Конрад Цузе
Деннис Ритчи
Джон Бэкус

5. Структурное программирование – это ...

проверка исходных данных и соответствующих им наборов эталонных решений.

процесс, посредством которого проверяется правильность программы.
дисциплина, обучающая языку программирования высокого уровня.
дисциплина, требующая, чтобы любая программа строилась из ограниченного набора типовых (базовых) управляющих структур и их композиций.

6. К базовым алгоритмическим структурам относятся:

ветвление;
блок;
переход;
цикл;
линейный алгоритм (цепочка);

7. На этапе проектирования нужно стремиться спланировать программу таким образом, чтобы

вносить изменения в нее было как можно сложнее
вносить изменения в нее было как можно проще
ее документация была как можно проще
ее документация была как можно сложнее

8. Каков минимальный набор управляющих структур, с помощью которого можно представить любой алгоритм?

- 1
- 2
- 3
- 4

9. Согласно теореме структурирования любой алгоритм можно представить в виде двух управляющих структур. Каких?

- «цепочка»
- «ветвление»
- цикл с предусловием
- цикл с постусловием
- цикл с конечным числом итераций

10. Характерной особенностью базовой алгоритмической структуры «цепочка» является наличие в ней...

- одного входа и двух выходов;
- одного входа и одного выхода;
- двух входов и двух выходов;
- двух входов и одного выхода;

11. Главная цель, к которой должен стремиться программист, при решении поставленной задачи, - получить

- легко читаемую программу, возможно, более простой структуры
- универсальную программу
- более сложную программу, чем созданная ранее
- «экзотическую» программу

12. Языками программирования являются:

- Лого
- Турбо Алгол
- Пролог
- Перл
- Симула

13. Следуя правилам программирования, нужно

- использовать в программе числа в явном виде
- избегать использования в программе чисел в явном виде
- константы называть сложными именами
- назначать переменным имена, скрывающие их смысл

14. Какие компоненты образуют алгоритмический язык?

- семантика

синтаксис
служебные слова
алфавит

15. В какой сфере язык Фортран нашёл наибольшее применение?

в сфере научно-технических расчётов
в торговой сфере
в программировании удалённого доступа
в сфере дизайна и анимации

16. Укажите основные преимущества алгоритмических языков по сравнению с машинными?

Формат предложений достаточно громоздок и неудобен для использования.

Алфавит алгоритмического языка значительно шире машинного.

Набор операций, допустимых для использования, не зависит от набора машинных операций.

Требуемые операции задаются с помощью общепринятых математических обозначений.

Алфавит алгоритмического языка значительно уже машинного.

17. Первым языком, в котором появился достаточный набор управляющих операторов, позволяющих писать программу без «go to», является:

Бейсик
Алгол
Паскаль
Кобол
Фортран

18. Данные делятся на:

константы
переменные
операции
операторы

19. Выражения алгоритмического языка предназначаются для выполнения необходимых вычислений и состоят из ...

констант, переменных, указателей функций, объединённых знаками операций.

констант и переменных.

переменных и указателей функций.

стандартных функций.

20. К критериям качества программы относятся:

эффективность
объем внешней памяти
документируемость
время выполнения

21. Под проектированием программы понимается
проверка исходных данных и соответствующих им наборов эталонных реакций

процесс, посредством которого проверяется правильность программы
создание технического задания
определение общей структуры и взаимодействия модулей

22. Спецификация интерфейсов – это
запись программы на языке программирования по готовому алгоритму

представление алгоритмов
определение способов взаимодействия подзадач
программирование задач

23. Процесс проектирования задачи является
линейным
итерационным
ветвящимся
кольцевым

24. Представление алгоритма решения задачи в виде последовательности подзадач называется

процедурной декомпозицией
спецификацией интерфейсов
структурным программированием
нисходящим тестированием

25. При структурном программировании организуется кодирование программы по принципу:

«сверху вниз»
«снизу-вверх»
«слева направо»
«справа налево»

4.2. Типовые вопросы, выносимые на зачет

1. Язык программирования как знаковая система
2. Этапы решения задач с помощью компьютера.

3. Классификация языков программирования в зависимости от степени детализации предписаний.
4. Генеалогия языков программирования высокого уровня.
5. Транслятор: компилятор, интерпретатор, конвертор.
6. Метаязыки.
7. Первое поколение языков. Фортран
8. Первое поколение языков. Алгол
9. Первое поколение языков. Кобол
10. Язык программирования ПЛ/1
11. Интерактивное программирование. Бейсик.
12. Структурированное программирование и его история.
13. Технология ООП.
14. Си- подобный синтаксис.
15. Язык Си. Состав языка.
16. Стандарт языка C/C++.
17. Компоненты алгоритмического языка. Примеры.
18. Основные стандартные функции в алгоритмических языках.
19. Лексика языка Си. Примеры.
20. Грамматика языка Си. Примеры.
21. C++ , как ЯПВУ объектно-ориентированной парадигмы.
22. Операторы ввода/вывода данных в C/C++.
23. Структурные и простые операторы C/C++. Примеры.
24. Типы данных. Классификация.
25. Простые типы данных в Си. Примеры.
26. Структурированные типы данных в Си. Примеры.

4.3. Типовые вопросы, выносимые на экзамен

1. Тип – массив. Примеры на Си/Си++
2. Тип – запись. Примеры Си/Си++.
3. Тип множество в ЯП ВУ.

4. Тип- строка в ЯП ВУ.
5. Тип- файл. Примеры в Си/Си++.
6. Тип класс в ЯП ВУ С++.
7. Процедуры передачи управления. Примеры.
8. Совместимость типов.
9. Подпрограммы. Процедуры и функции.
10. Глобальные и локальные переменные.
11. Структура исполняемой программы в оперативной памяти.
12. Примеры передачи двух основных видов параметров подпрограмм. Их отличие.
13. Рекурсивные подпрограммы. Примеры простейших функций.
14. Модули. Описание модулей. Использование модулей.
15. Стандартные модули С/С++.
16. Структура программы в Visual Studio(CODE::Blocks) на Си/Си++.
17. Пример модулей ввода-вывода в Visual Studio(CODE::Blocks) на Си.
18. Пример модулей ввода-вывода в Visual Studio(CODE::Blocks) на Си++.
19. Модуль математических функций в Visual Studio на Си/Си++ .
20. Оконные приложения в Visual Studio на Си++.
21. Стандартные библиотеки классов в С++.
22. Критерии качества программы.
23. Правила программирования.
24. Документирование программы.

Итоговое начисление баллов по дисциплине осуществляется в соответствии с разработанной и внедренной балльно-рейтинговой системой контроля и оценивания уровня знаний и внеучебной созидательной активности обучающихся.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ»**

(Приложение 2 к рабочей программе)

Направление подготовки: 09.03.04 Программная инженерия
Профиль: Проектирование и разработка программного обеспечения
Уровень высшего образования: бакалавриат
Форма обучения: очная

1. Общие положения

Целью изучения дисциплины является:

- развитие у студентов личностных качеств;
- формирование компетенций в соответствии с действующим образовательным стандартом по направлению подготовки : 09.03.04 «Программная инженерия».

Задачи дисциплины:

— подготовка обучающихся к *научно-исследовательской деятельности*: проведение научно-исследовательских мероприятий в рамках анализа качества программного продукта, определение возможных способов повышения качества, разработка комплекса мероприятий по повышению качества программного продукта;

— подготовка обучающихся к *проектной деятельности*: определение требований к программному продукту, разработка проекта программного продукта с использованием известных способов и методов проектирования, разработка проектной документации;

— подготовка обучающихся к *производственно-технологической деятельности*: реализация программного продукта на языке программирования с использованием известных технологий и методов программирования, тестирование, интеграция программных продуктов, повышение качества программного продукта, сопровождение программного продукта.

2. Указания по проведению практических занятий

Третий семестр

Тема: Введение в курс «Языки программирования»

Практическое занятие 1

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: диспут.

Цель работы: На примере ЯП ВУ изучить основные аспекты решения задач с использованием среды программирования.

Основные положения темы занятия:

1. ЯП как знаковая система.
2. Классификация ЯП.
3. Редактор
4. Транслятор. Виды трансляторов.

5. Компоновщик.
6. Библиотечные модули.

Вопросы для обсуждения:

1. Реализация языка программирования.
2. Современные системы программирования.
3. Основные компоненты среды программирования.
4. Современные парадигмы программирования.

Продолжительность занятия – 2 ч.

Тема: История создания ЯП ВУ. Состав ЯП

Практическое занятие 2

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: На примере одного из выбранных ЯП ВУ изучить основные компоненты языка программирования.

Основные положения темы занятия:

1. Алфавит ЯП
2. Лексема, выражение.
3. Оператор и операнд
4. Тип данных.

Вопросы для обсуждения:

1. Чем отличается ЯП от естественного языка.
2. Зачем нужны типы данных.
3. Синтаксис ЯП ВУ.
4. Семантика высокоуровневых языков программирования

Продолжительность занятия – 8 ч.

Тема: Структурное и модульное программирование.

Практическое занятие 3

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: дебаты (активное высказывание по позициям и обсуждение спорных вопросов).

Цель работы: Получение практических навыков в составлении алгоритмов решения различной сложности прикладных задач с последующей отладкой и тестированием написанного кода в соответствующей среде программирования.

Основные положения темы занятия:

1. Основные типы данных
2. Базовые алгоритмические структуры.
3. Процедуры и функции

Вопросы для обсуждения:

1. Простые и структурированные типы данных в ЯП ВУ.
2. Простые и структурные операторы ЯП ВУ.
3. Процедуры передачи управления.
4. Совместимость типов.
5. Подпрограммы. Процедуры и функции.
6. Глобальные и локальные переменные.
7. Структура исполняемой программы в оперативной памяти.
8. Виды параметров подпрограмм.

Продолжительность занятия – **10 ч.**

Тема: Основные принципы программирования на языках высокого уровня.

Практическое занятие 4

Вид практического занятия: *подготовка доклада.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Самостоятельная работа студентов по заранее выданным вопросам и обсуждение представленного материала на занятии в форме семинара.

Основные положения темы занятия:

1. Язык программирования и его детализация предписаний.
2. Правила программирования на ЯП ВУ.

Вопросы для обсуждения:

1. Классы языков
2. Императивная парадигма
3. Объектно-ориентированная парадигма
4. Структурные языки программирования
5. Критерии качества программы
6. Расширяемость программного кода

Продолжительность занятия – **16 ч.**

Четвёртый семестр

Тема: Языки и системы программирования ООП – парадигмы

Вид практического занятия: *подготовка доклада.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Самостоятельная работа студентов по заранее выданным вопросам и обсуждение представленного материала на занятии в форме семинара.

Основные положения темы занятия:

1. Вычислительная модель ООП.
2. Языки ООП.

Вопросы для обсуждения:

1. Основные понятия объектно-ориентированного программирования.
2. Современные среды, поддерживающие ООП.

3. Объекты.

4. Данные (поля) и методы.

Продолжительность занятия – 4 ч.

Тема: Технология объектно-ориентированного программирования

Практическое занятие 6

Вид практического занятия: подготовка доклада.

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Самостоятельная работа студентов по заранее выданным вопросам и обсуждение представленного материала на занятии в форме семинара.

Основные положения темы занятия:

1. Основные принципы объектно-ориентированного программирования.

2. Библиотеки классов.

Вопросы для обсуждения:

1. Инкапсуляция. Использование кода.

3. Наследование..

4. Полиморфизм.

5. Состав современных библиотек классов

Продолжительность занятия – 6 ч.

Тема: Стандартные библиотеки и их подпрограммы в C++.

Практическое занятие 6

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Научиться использовать стандартные модули среды программирования Visual Studio (CodeBlocks) для C++.

Основные положения темы занятия:

1. Создание модулей
2. Структура модулей
3. Содержание и состав стандартных модулей среды программирования Visual Studio (CodeBlocks) для C++.
4. Отличительные черты библиотечных функций для классов.

Вопросы для обсуждения:

1. Модули Visual Studio на Си++.
2. Библиотека STD
3. Преимущества библиотек классов
4. Модули ввода-вывода в C++.
5. Модули для работы со списками.

Продолжительность занятия – **16ч.**

Тема: Языки и системы функционального и логического программирования

Практическое занятие 8

Вид практического занятия: *подготовка доклада.*

Тема и содержание практического занятия:

Образовательные технологии: дебаты (активное высказывание по позициям и обсуждение спорных вопросов).

Цель работы: Самостоятельная работа студентов по заранее выданным вопросам и обсуждение представленного материала на занятии в форме семинара.

Основные положения темы занятия:

1. Вычислительная модель функционального программирования.
2. Вычислительная модель логического программирования

Вопросы для обсуждения:

1. Языки и среды функционального программирования
2. Языки и среды логического программирования
3. Мульти парадигмальные современные ЯП.

Продолжительность занятия – **6 ч.**

3. Указания по проведению лабораторного практикума

Не предусмотрен учебным планом.

4. Указания по проведению самостоятельной работы студентов

Цель самостоятельной работы: подготовить бакалавров к самостоятельному научному творчеству.

Задачи самостоятельной работы:

- расширить знания по современным языкам и средам программирования;
- овладеть технологией модульного и структурного программирования;
- систематизировать знания, подходы, методы по решению прикладных задач пользователя с помощью ЭВМ

Виды самостоятельной работы представлены в таблице 1.

Таблица 1

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1	Тема 1 Введение в курс «Языки программирования»	Самостоятельное изучение тем, подготовка рефератов. Вопросы, выносимые на самостоятельное изучение: 1. История развития языков программирования высокого уровня 2. Императивное программирование 3. Языки высокого и низкого уровня: отличительные особенности. Примерная тематика рефератов: 1. Язык программирования как знаковая система. 2. Первые ЭВМ. Язык Планкалкюль. 3. Язык программирования Фортран. 4. Язык программирования Алгол.
2	Тема 2. История создания ЯП ВУ. Состав языка.	Самостоятельное изучение тем, создание презентаций. Вопросы, выносимые на самостоятельное изучение: 1. Реализация языка. 2. Задачи, решаемые с помощью систем программирования. Примерная тематика докладов с презентацией: 1. Формализация задач объективного мира.

		<ol style="list-style-type: none"> 2. Конвертор – тип транслятора. 3. Компановщик: назначение, примеры. 4. Современные среды программирования на ЯВУ.
3	Тема 3. Структурное и модульное программирование.	<p>Самостоятельное изучение тем, создание презентаций.</p> <p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Иерархическое дерево развития ЯВУ 2. Область применения современных языков программирования высокого уровня <p>Примерная тематика докладов для презентаций:</p> <ol style="list-style-type: none"> 1. Язык программирования VisSim 2. Язык программирования Perl 3. Язык программирования РЕФАЛ 4. Язык программирования Python 5. Язык программирования Ruby 6. Язык программирования Scala 7. Язык программирования Оберон 8. Язык программирования Симула 9. Язык программирования Ladder Diagram (LD)
4	Тема 4. Основные принципы программирования на языках высокого уровня.	<p>Самостоятельное изучение тем, подготовка рефератов.</p> <p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Структура и состав современных ЯВУ. 2. Область применения современных ЯП. <p>Примерная тематика рефератов:</p> <ol style="list-style-type: none"> 1. Синтаксис и семантика языков программирования. 2. Широта использования современных языков и систем программирования. 3. Основные парадигмы и концепции программирования в свете ЯВУ. 4. Основные правила программирования структурного подхода в программировании. 5. Язык программирования АДА.
5	Тема 5. Языки и системы программирования ООП – парадигмы	<p>Самостоятельное изучение тем, создание презентаций, подготовка реферата.</p> <p>Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Современные объектно-ориентированные ЯП. 2. Среда программирования ООП концепции. <p>Примерная тематика рефератов</p> <ol style="list-style-type: none"> 1. Язык программирования C#. 2. Язык программирования C++. 3. Язык программирования Java. 1. Язык программирования Objectiv-C

	<p>Тема 6. Технология объектно-ориентированного программирования.</p>	<p>Самостоятельное изучение тем, создание презентаций, подготовка реферата. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Преимущества ООП технологий. 2. Инициализация элементов класса. 3. Оконный интерфейс. <p>Примерная тематика рефератов:</p> <ol style="list-style-type: none"> 1. Основные принципы объектно-ориентированного программирования. 2. Вычислительная модель в современных ООП-средах. 3. Языки ООП назначения. 4. Конструкторы и деструкторы.
6	<p>Тема 7. Стандартные библиотеки и их подпрограммы в С++</p>	<p>Самостоятельное изучение тем, создание презентаций, подготовка реферата. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Библиотеки современных сред программирования. 2. Состав стандартных библиотек сред программирования. <p>Примерная тематика рефератов</p> <ol style="list-style-type: none"> 1. Библиотека Math среды программирования на С/ С++. 2. Библиотека Syslib 3. Библиотеки работы с графикой в популярных средах программирования. 4. Библиотечные функции работы с элементами динамических структур в С++ 5. Библиотеки, обеспечивающие оконный интерфейс..
8	<p>Тема 8 Языки и системы функционального и логического программирования</p>	<p>Самостоятельное изучение тем, создание презентаций, подготовка реферата. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Интересы современных пользователей сред программирования на ЯВУ 2. Экспертные системы и их программная основа. 3. Современные среды программирования, поддерживающие функциональность. <p>Примерная тематика рефератов</p> <ol style="list-style-type: none"> 1. Язык программирования Prolog. 2. Язык программирования Haskell. 3. Язык программирования Clean. 4. Задачи, решаемые функциональным программированием. 5. Задачи, решаемые в средах логического программирования.

5. Указания по проведению контрольных работ

5.1. Требования к структуре

Структура контрольной работы должна способствовать раскрытию темы: иметь титульный лист, содержание, введение, основную часть, заключение, список литературы.

5.2. Требования к содержанию (основной части)

1. Во введении обосновывается актуальность темы, определяется цель работы, задачи и методы исследования.

2. При определении целей и задач исследования необходимо правильно их формулировать. Так, в качестве цели не следует употреблять глагол «сделать». Правильно будет использовать глаголы: «раскрыть», «определить», «установить», «показать», «выявить» и т.д.

3. Основная часть работы включает 2 - 4 вопроса, каждый из которых посвящается решению задач, сформулированных во введении, и заканчивается констатацией итогов.

4. Приветствуется иллюстрация содержания работы таблицами, графическим материалом (рисунками, схемами и т.п.).

5. Необходимо давать ссылки на используемую Вами литературу.

6. Заключение должно содержать сделанные автором работы выводы, итоги исследования.

7. Вслед за заключением идет список литературы, который должен быть составлен в соответствии с установленными требованиями. Если в работе имеются приложения, они оформляются на отдельных листах, и должны быть соответственно пронумерованы.

5.3. Требования к оформлению

Объём контрольной работы – 10-15 страниц формата А4, напечатанного с одной стороны текста (1,5 интервал, шрифт Times New Roman).

6. Перечень основной и дополнительной учебной литературы

Основная литература:

1. С++ как второй язык в обучении приемам и технологиям программирования: учебное пособие / Я.М. Русанова. - Ростов-на-Дону: Издательство ЮФУ, 2010. - 200 с. ISBN 978-5-9275-0749-8

<http://znanium.com/catalog.php?item=bookinfo&book=550811>

2. Бедердинова О. И. Программирование на языках высокого уровня: учеб. пособие / О.И. Бедердинова, Т.А. Минеева, Ю.А. Водовозова. — Москва:

ИНФРА-М, 2019. — 159 с. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1044396>

3. Структуры и алгоритмы обработки данных: Учебное пособие / В.Д. Колдаев. - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014. - 296 с.: 60x90 1/16. - (Высшее образование: Бакалавриат). (переплет) ISBN 978-5-369-01264-2, 500 экз. <http://www.znanium.com/catalog.php>

4. Белов В.В., Чистякова В.И. Алгоритмы и структуры данных: Учебник М.: КУРС, НИЦ ИНФРА-М, 2016. - 240 с. ISBN 978-5-906818-25-6. / ЭБС «Знаниум».

Дополнительная литература:

1. Колдаев В.Д. Численные методы и программирование: учеб. пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 336 с.: ил. — (Профессиональное образование).

<http://znanium.com/catalog.php?item=bookinfo&book=672965>

2. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 517 с. + Доп. материалы [Электронный ресурс; Режим доступа <http://www.znanium.com>]. — (Высшее образование). — www.dx.doi.org/10.12737/23113.

<http://znanium.com/catalog.php?item=bookinfo&book=562914>

3. Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке С++: Учебное пособие. - 1. - Москва; Москва: Издательский Дом "ФОРУМ": ООО "Научно-издательский центр ИНФРА-М", 2018. - 512 с. URL: <http://znanium.com/go.php?id=918098>

4. Языки программирования: учеб. пособие / О.Л. Голицына, Т.Л.Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — М.: ФОРУМ: ИНФРА-М, 2018. — 399 с. URL: <http://znanium.com/bookread2.php?book=973007>

5. Галимов, Р. Р. Программирование в среде Visual С++ с использованием библиотеки МФС: метод. указания к лаб. работам / А. И. Сарайкин, Оренбургский гос. унт, Р. Р. Галимов. — Оренбург: ОГУ, 2014 <http://rucont.ru/efd/271378?cldren=0>

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

<http://www.znanium.com/> – электронно-библиотечная система Znanium;

<http://www.e.lanbook.com/> – ЭБС Издательства «ЛАНЬ»;

<http://www.rucont.ru/> – электронно-библиотечная система;

<http://www.biblioclub.ru/> – университетская библиотека онлайн.

8. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в Приложении 2 к настоящей Рабочей программе.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения:

Прикладные программы CODE::Blocks, Visual Studio, ППП ONLYOFFICE.

Информационные справочные системы:

1. Электронные ресурсы образовательной среды Университета.
2. Рабочая программа и методическое обеспечение по дисциплине «Языки программирования высокого уровня»