



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

«__» _____ 2023 г.

ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ
«ТЕХНОЛОГИИ И СРЕДЫ ПРОГРАММИРОВАНИЯ»

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 профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор: к.т.н., с.н.с. Парафейников И.В. Рабочая программа дисциплины: Технологии и среды программирования. – Королев МО: «Технологический Университет», 2023

Рецензент: д.т.н., профессор Артюшенко В.М.

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 (переподтверждения)	2023	2024	2025	2026
Номер и дата протокола заседания кафедры	№ 7 от 03.04.2023			

Рабочая программа согласована:

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

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переподтверждения)	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 ВО

Целью изучения дисциплины является:

- 1) ознакомление с принципами, базовыми концепциями технологий программирования, выступающими как составная часть технологии разработки объектов профессиональной деятельности;
- 2) формирование и развитие компетенций, знаний, практических навыков и умений, обеспечивающих разработку средств реализации информационных технологий (в первую очередь информационных, алгоритмических и программных);
- 3) практическое освоение интегрированных сред изучаемого алгоритмического языка высокого уровня;
- 4) изучение основных этапов и принципов создания программного продукта, конструктивных компонентов и структур данных, способов их представления и обработки;
- 5) изучение методов обработки исключений, ошибок и подходов к отладки программных продуктов.

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:

- Изучение базовых концепций и технологий программирования, выступающих, как составная часть технологии разработки объектов профессиональной деятельности (программных продуктов);
- Получение студентами базовых знания об организации процессов управления разработкой и отладкой программных продуктов командой разработчиков и их взаимодействия с постановщиками и заказчиками программного продукта;

- Получение студентами умений и навыков использования интегрированных сред разработки, применяемых для решения практических задач создания программных продуктов.

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 «Технологии и среды программирования» 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 и виды учебной работы

Общая трудоемкость дисциплины составляет 12 зачетных единиц, 432 часа.

Таблица 1

Виды занятий	Всего часов	Семестр второй	Семестр третий	Семестр четвертый
Общая трудоемкость	432	144	144	144
Аудиторные занятия	192	64	48	80
Лекции (Л)	80	32	16	32
Практические занятия (ПЗ)	112	32	32	48
Лабораторные работы (ЛР)	-	-	-	-
Практическая подготовка	-	-	-	-
Самостоятельная работа	240	80	96	64
Курсовые работы (проекты)	+	-	-	+
Расчетно-графические работы	-	-	-	-
Контрольная работа	+	+	+	+
Текущий контроль знаний	Тест	Тест	Тест	Тест
Вид итогового контроля	Зачёт/ Зачёт с оценкой/ Экзамен	Зачёт	Зачёт с оценкой	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
Тема 1. Основные технологии разработки программного обеспечения	8	8	2	ОПК-2
Тема 2. Оценка качества процессов создания ПО. Подходы к управлению процессами разработки ПО	8	8	2	ОПК-3
Тема 3. Среды программирования Code::Blocks и Visual Studio Code	8	8	2	ОПК-6 ОПК-7
Тема 4. Среды программирования JetBrains PyCharm и Jupyter Notebook	8	8	2	ОПК-6 ОПК-7
Тема 5. Среда программирования Eclipse	16	32	12	ОПК-6 ОПК-7
Тема 6. Среды программирования NetBeans и PHPStorm	32	48	16	ОПК-6 ОПК-7
Итого:	80	112	36	

4.2. Содержание тем дисциплины

Тема 1. Основные технологии разработки программного обеспечения.

- 1.1. Этапы развития программирования.
- 1.2. Этапы жизненного цикла программного обеспечения.
- 1.3. Подходы к организации создания и использования ПО.
- 1.4. Принципы разработки программных систем и приемы обеспечения технологичности программного обеспечения.

Тема 2. Оценка качества процессов создания ПО. Подходы к управлению процессами разработки ПО.

- 2.1. Эволюция моделей жизненного цикла программного обеспечения.
- 2.2. Оценка качества процессов создания программного обеспечения.
- 2.3. Приёмы обеспечения технологичности программного обеспечения.
- 2.4. Подходы к управлению процессами разработки ПО.

Тема 3. Среды программирования Code::Blocks и Visual Studio Code.

- 3.1. Состав среды программирования Code::Blocks.
- 3.2. Сборка, запуск и отладка программы
- 3.3. Настройки редактора и консоли
- 3.4. Установка и настройка среды программирования Visual Studio Code.
- 3.5. Интерфейс Visual Studio Code.
- 3.6. Создание проекта, сборка, запуск и отладка программы.
- 3.7. Горячие клавиши Visual Studio Code.

Тема 4. Среды программирования JetBrains PyCharm и Jupyter Notebook.

- 4.1. Состав среды программирования PyCharm.
- 4.2. Создание и активация виртуального окружения IDE PyCharm
- 4.3. Создание и запуск проекта. Особенности отладки и тестирования с IDE JetBrains PyCharm.
- 4.4. Как устроен и работает Jupyter Notebook
- 4.5. Виджеты Jupyter Notebook
- 4.6. Основные элементы интерфейса Jupyter Notebook

Тема 5. Среда программирования Eclipse.

- 5.1. Состав среды программирования Eclipse.
- 5.2. Создание и активация виртуального окружения IDE Eclipse.
- 5.3. Создание и запуск проекта. Особенности отладки и тестирования с IDE Eclipse.
- 5.4. Настройка среды программирования Visual Studio Code для Java.
- 5.5. Создание и запуск проекта. Особенности отладки и тестирования с IDE VS Code.
- 5.6. Популярные фреймворки.

Тема 6. Среда программирования NetBeans и PHPStorm.

- 6.1. Состав среды программирования NetBeans.
- 6.2. Создание и активация виртуального окружения IDE NetBeans
- 6.3. Создание и запуск проекта. Особенности отладки и тестирования с IDE NetBeans.
- 6.4. Как устроен и работает PHPStorm
- 6.5. Создание и запуск проекта. Особенности отладки и тестирования с IDE PHPStorm
- 6.6. Основные элементы интерфейса PHPStorm

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

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в Приложении 1.

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

Основная литература:

1. Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.: КУРС, НИЦ ИНФРА-М, 2017. - 240 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-906818-25-6

<http://znanium.com/catalog.php?item=bookinfo&book=766771>

2. Слабнов В. Д. Программирование на С++ / В. Д. Слабнов; В.Д. Слабнов. - Казань: Познание, 2012. - 136 с. - ISBN 978-5-8399-0386-9.

URL: <http://biblioclub.ru/index.php?page=book&id=364222>

3. Основы алгоритмизации и программирования на Python: учеб. пособие / С.Р. Гуриков. — М.: ФОРУМ: ИНФРА-М, 2017. — 343 с. — (Высшее образование: Бакалавриат).

<http://znanium.com/catalog.php?item=bookinfo&book=924699>

4. Карепова, Е.Д. Основы многопоточного и параллельного программирования: учебное пособие / Е.Д. Карепова; Сибирский Федеральный университет, Федеральное государственное бюджетное учреждение науки «Институт вычислительного моделирования Сибирского отделения Российской академии наук», Сибирский научно-образовательный центр суперкомпьютерных технологий. – Красноярск: СФУ, 2016. – 355 с.: ил. – Режим доступа: по подписке.

– URL: <http://biblioclub.ru/index.php?page=book&id=497217>

Дополнительная литература:

1. Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 400 с. — (Высшее образование: Бакалавриат).

<http://znanium.com/catalog.php?item=bookinfo&book=924760>

2. Горелов С.В. Современные технологии программирования: разработка Windows-приложений на языке С#: учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика» (09.03.03 — для бакалавров, 09.04.03 — для магистров): в 2 т.: [16+] / С.В. Горелов; под науч. ред. П.Б. Лукьянова; Финансовый университет при Правительстве Российской Федерации. – Москва: Прометей, 2019. – Том 1. – 363 с.: ил. – Режим доступа: по подписке. –

URL: <http://biblioclub.ru/index.php?page=book&id=576037>

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

Интернет-ресурсы:

- <http://www.codeblocks.org/>
- <https://code.visualstudio.com/>
- <http://www.cplusplus.com/> - Сеть ресурсов, посвященных языку С++
- <http://dia-installer.de/>
- <https://www.jetbrains.com/> - Сайт компании-разработчика JetBrains
- <https://notepad-plus-plus.org/>
- <https://www.python.org/> – Сайт разработчиков Python
- <https://jupyter.org/>
- <https://colab.research.google.com/>
- <http://pythontutor.ru/> - Курс по программированию на языке Python
- <https://netbeans.apache.org/front/main/download/index.html>

- <https://www.netbeans.info/products/ide/features.html>
- <https://www.jetbrains.com/ru-ru/phpstorm/>
- <https://www.jetbrains.com/ru-ru/phpstorm/documentation/>
- <https://www.eclipse.org/downloads/>
- https://www.tutorialspoint.com/eclipse/eclipse_quick_guide.htm
- <https://stepik.org> Stepik — образовательная платформа и конструктор онлайн-курсов.
- <http://www.znaniium.com/> - Электронная библиотека издательства "Инфра-М"

9. Методические указания для обучающихся по освоению дисциплины

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

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

Перечень программного обеспечения: Dia, Double Commander, NotePad++, Microsoft Office или свободно распространяемые аналоги, VS Code, GCC + MinGW, Python 3, PyCharm, Jupyter Notebook, Eclipse, PHPStorm, NetBeans.

Информационные справочные системы: Электронные ресурсы образовательной среды Университета

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

Лекционные занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран), доска, комплект маркеров;
- комплект электронных презентаций;
- рабочие места студентов, оснащенные компьютерами с доступом (управляемым с компьютера преподавателя) в Интернет к почтовым серверам и к адресам, приведенным в разделе 8 и к общей сетевой папке группы.

Практические занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран), доска, комплект маркеров;
- комплект электронных презентаций;
- рабочие места студентов, оснащенные компьютерами с доступом (управляемым с компьютера преподавателя) в Интернет к почтовым серверам, к адресам, приведенным в разделе 8 и к общей сетевой папке группы.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ
«ТЕХНОЛОГИИ И СРЕДЫ ПРОГРАММИРОВАНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Профиль: Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

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 (или ее части), обучающийся должен:		
				Необходимые знания	Необходимые умения	Трудовые действия
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 1	Знает принципы работы современных информационных технологий и программных средств, в том числе отечественного производства.	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;	Применяет современные информационные технологии и программные средства, в том числе, отечественного производства при решении задач профессиональной деятельности;
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	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	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;	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

				информационн ой безопасности		информационн ой безопасности; Анализирует информационн о- коммуникацио нных технологий и с учетом основных требований информационн ой 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 3 Тема 4 Тема 5 Тема 6	Знает 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;	Применяет 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,	Тема 3 Тема 4 Тема 5 Тема 6	Знает основные концепции, принципы, теории и факты,	Использует основные концепции, принципы, теории и факты, связанные с	Применяет в практической деятельности основные концепции, принципы,

	принципы, теории и факты, связанные с информатикой.		связанные с информатикой	информатикой для решения практических задач;	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-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 <u>не сформирована</u>) – 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> <ol style="list-style-type: none"> 1. Соответствие содержания контрольной работы заявленной тематике (1 балл). 2. Качество источников и их количество при подготовке работы (1 балл). 3. Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4. Качество самой представленной работы (1 балл). 5. Использование специализированного программного обеспечения (1 балл). 6. Оригинальность подхода и всестороннее раскрытие выбранной тематике (1 балл). <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

3.2 Примерная тематика заданий контрольной работы

1. Разработать модуль / библиотеку для операций “длинной арифметики, то есть сложения, умножения, вычитания для чисел размером в 1000 цифр.
2. Разработать модуль / библиотеку для реализации вектора через указатели и массивы
3. Разработать программу построения графиков функций (функция задается в текстовом файле с использованием действий «+», «-», «*», «/», «^» степень). Скобки не используются
Например, $y=5*x^2+3$ или $F=6+5/E^2$. На экране должны быть изображены оси с подписями или сетка, график и надпись –формула.
4. Разработать программу построения графиков функций (функция задается в текстовом файле с использованием действий «+», «-», «*», «/», «^» степень). Скобки не используются
Например, $y=10*x^3+7$ или $F=6+5/E^2$. На экране должны быть изображены оси с подписями или сетка, график и надпись –формула.
5. Разработать класс «Очередь» на основе указателей и отразить визуально операции с очередью.
6. Написать программу построения статистической диаграммы встречаемости символов в текстовом файле.
7. Написать программу чтения таблицы из Excel-файла и формирования HTML-страницы.
8. Написать программу чтения списка из WORD и вывода в HTML.
9. Написать программу для размещения на прямоугольном листе с заданным размером деталей прямоугольной и круглой формы. Детали не должны пересекаться. Программа должна выводить в текстовый файл отчет по следующей форме, например
Лист-длина, ширина
Прямоугольник – лев, низ, длина, ширина
Круг – лев, низ, радиус
Коэффициент использования площади = (вычисляется как отношение суммарной площади фигур к площади листа)
10. Разработать программу ведения англо-русского словаря. Организовать структуру для возможности быстрого (двоичного) поиска
11. Разработать приложение для построения графика функции $y=a*\sin(x)+b$ в координатных осях. a и b считываются из текстового файла.
12. Разработать приложение для построения графика полинома n -й степени. Коэффициенты полинома считываются из csv-файла
13. Разработать приложение для нахождения решения системы $3x$ линейных уравнений с 3 мя неизвестными. Коэффициенты считываются из csv-файла
14. Текстовый файл содержит слова, разделенные одним или несколькими пробелами. Разработать приложение для построения отсортированного словаря используемых слов. Вывести словарь в файл
15. Разработать приложение для нахождения обратной матрицы для матрицы $3*3$. Коэффициенты матрицы считываются из текстового файла. Коэффициенты обратной матрицы выводятся в html-файл.

16. Разработать приложение для демонстрации работы с АТД "Стек".
Использовать модуль.

17. Разработать приложение для демонстрации работы с АТД "Очередь".
Использовать модуль.

18. Разработать приложение для демонстрации работы с АТД "Дек".
Использовать модуль.

19. Разработать приложение для демонстрации работы с АТД "Множество".
Использовать модуль.

20. Разработать приложение для демонстрации работы с АТД "Односвязный список".
Использовать модуль.

21. Разработать приложение для демонстрации работы с АТД "Двусвязный список".
Использовать модуль.

22. Разработать приложение для демонстрации работы с АТД "Очередь с приоритетом".
Использовать модуль.

23. Разработать приложение для демонстрации работы с АТД "Кольцевой список".
Использовать модуль.

24. Дан текстовый файл, содержащий ключевые слова. Вывести список всех файлов "*.pas" в заданном каталоге, содержащий хотя бы одно из ключевых слов.

25. Дан текстовый файл, содержащий ключевые слова. Вывести список всех файлов "*.pas" в заданном каталоге, содержащий все ключевые слова.

26. Запись "Студент" содержит поля "Фамилия" "Группа" "СтудБилет".
Разработать приложение для ведения файла студентов: добавление, удаление, поиск по номеру билета.

27. Запись "Группа" содержит поля "Номер" "Курс" "ЧислоСтудентов".
Разработать приложение для ведения файла студентов: изменение числа студентов, перевод на следующий курс, поиск по номеру группы.

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

Формой контроля знаний по дисциплине «Технологии и среды программирования» являются шесть текущих аттестаций в форме тестов и итоговая аттестация в формах зачета во втором семестре, зачета с оценкой в третьем семестре, экзамена в четвертом семестре.

Неделя текущего контроля	Вид оценочного средства	Код компетенции, 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 15-16	Тестирование	ОПК-2 ОПК-3 ОПК-6 ОПК-7	15 вопросов	Компьютерное тестирование; время, отведенное на процедуру - 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - 0 Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%.

						Отлично – от 90%. Максимальная оценка – 5 баллов
В соответствии с КУГ	Зачет	ОПК-2 ОПК-3 ОПК-6 ОПК-7	1 вопроса и 1 задача	Зачет проводится в письменной форме, путем ответа на вопросы. Время, отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения экзамена	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	Зачет с оценкой	ОПК-2 ОПК-3 ОПК-6 ОПК-7	1 вопрос и 1 задача	Зачет с оценкой проводится в письменной форме, путем ответа на вопросы. Время, отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения экзамена	Критерии оценки: « Отлично »: 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 »: • знание основных понятий предмета; • умение использовать и применять полученные знания на практике;

						<ul style="list-style-type: none"> • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета • неправильно решено практическое задание <p>«Удовлетворительно»: демонстрирует частичные знания по темам дисциплин; незнание неумение использовать и применять полученные знания на практике; не работал на практических занятиях;</p> <p>Неудовлетворительно»: 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.</p>
В соответствии с КУГ	Экзамен	ОПК-2 ОПК-3 ОПК-6 ОПК-7	2 вопроса и 1 задача	Экзамен проводится в письменной форме, путем ответа на вопросы. Время, отведенное на процедуру – 45 минут.	Результаты предоставляются в день проведения экзамена	<p>Критерии оценки:</p> <p>«Отлично»: 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.</p> <p>«Хорошо»: • знание основных понятий предмета; • умение использовать и применять</p>

						<p>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 • неправильно решено практическое задание <p>«Удовлетворительно»: демонстрирует частичные знания по темам дисциплин; незнание неумение использовать и применять полученные знания на практике; не работал на практических занятиях;</p> <p>Неудовлетворительно»: 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.</p>
--	--	--	--	--	--	---

4.1. Типовые вопросы, выносимые на тестирование

Процессор выполняет команды алгоритма, записанные ...

1. На алгоритмическом языке
2. На командном языке
3. В виде блок-схемы
4. На естественном языке
5. На машинном языке (в двоичном коде)

С фамилией какого из древних ученых связано происхождение слова «алгоритм»?

1. Евклид
2. Аль-Хорезми
3. Аль-Хайсама
4. Аль-Коши

5. Пифагор

Автором самого древнего алгоритма считается...

1. Евклид
2. Пифагор
3. Декарт
4. Аль-Хорезми
5. Аль-Коши

Какой из документов является алгоритмом?

1. Классный журнал
2. Расписание звонков
3. Правила техники безопасности
4. Инструкция по получению денег в банкомате
5. Список класса

Для представления алгоритма в графическом виде используют...

1. Геометрические фигуры
2. Линии, точки
3. Формулы
4. Все ответы правильны
5. Графики функции

Свойство алгоритма «дискретность» означает:

1. Предлагаемые действия должны быть понятными и единственно возможными
2. Способность алгоритма давать правильные результаты решения задач
3. Пригодность алгоритма для решения однотипных задач
4. Решение задач должно быть получено за определенное число шагов
5. Непрерывность алгоритмического процесса

Алгоритм — это...

1. последовательность команд для ЭВМ
2. совокупность требований к программе
3. набор определений и правил для исполнителя
4. отдельные указания исполнителю выполнить некоторые законченные действия
5. совокупность понятных и точных указаний о том, какие действия и в какой последовательности выполнять для решения любой задачи из заданного класса за конечное число шагов

Последовательность нескольких команд алгоритма, выполняемых одна за другой, называется...

1. Командой
2. Программа
3. Служебные слова
4. Серия
5. Система программ

Аргументами называются величины...

1. не являющиеся исходными данными для алгоритма

2. являющиеся результатами для алгоритма
3. используемые для обозначения
4. являющиеся исходными данными для алгоритма
5. являющиеся заголовком для алгоритма

Алгоритмы, целиком используемые в составе других алгоритмов, называются...

1. Линейными
2. Определенными
3. Разветвляющимися
4. Вспомогательными
5. Циклическими

В каком году разработан алгоритм Брезенхейма

1. 1983
2. 1957
3. 1978
4. 1965

Какой алгоритм заливки произвольной области с отверстиями более сложен

1. Рекурсивный
2. Рекурсивный на основе серий пикселей
3. Алгоритм с предварительным выделением границы
4. Примерно равны

Какой алгоритм заливки произвольной области с отверстиями выигрывает в скорости при многократной заливке

1. Рекурсивный
2. Рекурсивный на основе серий пикселей
3. Алгоритм с предварительным выделением границы
4. Примерно равны

Какая из структур данных используется для оптимизации скорости выполнения расщепления полигона на два по хорде

1. Двусвязный список
2. Двусвязный циклический список
3. Циклический список
4. Динамический массив

Какая структура данных используется для оптимизации скорости выполнения алгоритмов обработки полигонов

1. Двусвязный список
2. Двусвязный циклический список
3. Циклический список
4. Динамический массив

Какой из методов имеет аппаратную реализацию

1. Метод трассировки лучей
2. Метод z-буфера
3. Метод двоичного разбиения пространства
4. Метод плавающего горизонта

Какой из алгоритмов позволяет отображать графики функций от двух переменных

1. Метод трассировки лучей
2. Метод z-буфера
3. Метод двоичного разбиения пространства
4. Метод плавающего горизонта

Циклический алгоритм используется при вычислении ...

1. суммы всех чисел от 1 до 100
2. площади трапеции
3. корня квадратного уравнения
4. суммы двух чисел, введенных с клавиатуры
5. среднего арифметического всех двухзначных чисел

Разветвляющийся алгоритм – это ...

1. присутствие в алгоритме хотя бы одного условия
2. набор команд, которые выполняются последовательно друг за другом
3. многократное выполнение одних и тех же действий
4. алгоритм, использующий подпрограммы

Свойство алгоритма «дискретность» обозначает ...

1. что команды должны следовать друг за другом
2. что каждая команда должна быть описана в расчёте на конкретного исполнителя
3. разбиение алгоритма на конечное число простых шагов
4. обязательное наличие завершающих инструкций
5. последовательность выполнения команд алгоритма

Свойство алгоритма, указывающее, что каждое правило алгоритма должно быть чётким,

1. однозначным и не оставлять места для произвола, называется ...
2. дискретность
3. понятность
4. определённость
5. результативность
6. массовость

Понятность, массовость – это свойства ...

1. алгоритма
2. информации
3. кодирования
4. кибернетики
5. программы

Языками высокого уровня являются ...

1. машинный язык
2. язык ассемблера
3. процедурный язык
4. объектно-ориентированный язык
5. логический язык

Ошибки, которые обнаруживаются транслятором: ...

1. неверное образование имён переменных
2. пропуск знака пунктуации
3. неверное написание служебных слов
4. несогласованность скобок
5. неверное определение порядка арифметических действий
6. неполный учёт возможных условий
7. неправильно составленный алгоритм решения задачи

Верными утверждениями являются: ...

1. различают два вида трансляторов: компиляторы и интерпретаторы
2. различают два вида компиляторов: трансляторы и интерпретаторы
3. различают два вида интерпретаторов: компиляторы и трансляторы
4. транслятор – программа, переводящая текст программы на языке высокого уровня в эквивалентную программу на машинном языке
6. после того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны

4.2. Типовые вопросы, выносимые на зачёт

- 1) Дать определения следующим понятиям, привести примеры: а) алгоритм; б) алгоритмический язык; в) язык программирования; г) система программирования; д) синтаксис и семантика.
- 2) Дать определения следующим понятиям, привести примеры: а) машинный код; б) ассемблер и язык ассемблера; в) язык высокого уровня; г) парадигма программирования; д) декларативный язык; е) императивный язык.
- 3) Дать определения следующим понятиям, привести примеры: а) процедурное/функциональное программирование; б) структурное программирование; в) подпрограмма (функция); г) компилятор; д) интерпретатор.
- 4) Дать определения следующим понятиям, привести примеры: а) интегрированная среда разработки и её состав; б) базовые управляющие конструкции в языке программирования и их назначения; в) принципы структурного программирования.
- 5) Схема классической системы программирования?
- 6) Охарактеризуйте основные этапы развития технологии программирования.
- 7) Что понимается под стихийным программированием?
- 8) Чем характеризуется структурный подход к программированию?
- 9) Что такое модульное программирование?
- 10) В чем заключается объектный подход к программированию?
- 11) Что такое инкапсуляция, наследование, полиморфизм?
- 12) Сущность компонентного подхода в программировании?
- 13) Что такое COM-технология и DCOM-технология?
- 14) Что такое технология CORBA?
- 15) Что такое CASE-технология?

4.3. Типовые вопросы, выносимые на зачет с оценкой

- 1) Охарактеризуйте проблемы разработки сложных программных систем?
- 2) Какие специфические особенности разработки программных систем вы знаете?
- 3) В чем заключается сущность блочно-иерархического подхода к созданию системы?
- 4) Что такое декомпозиция системы?
- 5) Сущность метода пошаговой детализации?
- 6) Что понимается под локальной оптимизацией?
- 7) Что лежит в основе блочно-иерархического подхода к созданию программных систем?
- 8) Какие существуют подходы к организации создания и использованию программных систем?
- 9) Перечислите этапы процесса разработки программных систем?
- 10) Какие этапы разработки программных систем регламентируются ГОСТ19.102–77?
- 11) Какие модели жизненного цикла программного обеспечения существуют?
- 12) Что понимается под каскадной моделью жизненного цикла ПО?
- 13) Что понимается под моделью жизненного цикла ПО с промежуточным контролем?
- 14) Что понимается под спиральной моделью жизненного цикла ПО?
- 15) Что понимается под оценкой качества процессов создания ПО?
- 16) Какие группы и показатели качества ПО вы знаете?
- 17) Какие качественные характеристики ПО необходимо обеспечивать (ГОСТ Р ИСО 9126)?
- 18) Что понимается под технологичностью ПО и какие концепции положены в её основы?
- 19) Чем определяется технологичность ПО?
- 20) Что понимается под процедурной декомпозицией разрабатываемого ПО?
- 21) Что понимается под объектной декомпозицией разрабатываемого ПО?

4.4. Типовые вопросы, выносимые на экзамен

- 1) Что такое модуль и какие требования к нему предъявляются?
- 2) Какие способы обеспечения высокого уровня технологичности разработки ПО вы знаете?
- 3) Что такое сцепление модулей, какие типы сцепления модулей существуют?
- 4) Что такое связность модулей, какие виды связности модулей существуют?
- 5) Какие библиотеки ресурсов существуют?
- 6) Что понимается под нисходящей и восходящей разработкой ПО?
- 7) Что такое Agile?
- 8) Четыре ценности Agile?
- 9) Дайте характеристики принципам Agile?
- 10) Что такое Scrum?
- 11) Сущность принципов методологии Scrum?

- 12) На чём основан Scrum. В чём отличие Scrum от Agile?
- 13) Основная терминология Scrum?
- 14) Что такое спринт в Scrum?
- 15) Содержание цикла работы Scrum?
- 16) Каким образом определяется приоритетность работ в Scrum?
- 17) Как оценить успешность внедрения методологии Agile?
- 18) Что такое Jira, для чего она используется и какие задачи решаются с её помощью?
- 19) Среда программирования Code::Blocks?
- 20) Среда программирования PyCharm?
- 21) Среда программирования Visual Studio Code?
- 22) Среда программирования Jupyter Notebook?
- 23) Среда программирования Eclipse?
- 24) Среда программирования NetBeans?
- 25) Среда программирования PHPStorm?

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
«ТЕХНОЛОГИИ И СРЕДЫ ПРОГРАММИРОВАНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Профиль: Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

1. Общие положения

Цель дисциплины:

1. ознакомление с принципами, базовыми концепциями технологий программирования, выступающими как составная часть технологии разработки объектов профессиональной деятельности;
2. формирование и развитие компетенций, знаний, практических навыков и умений, обеспечивающих разработку средств реализации информационных технологий (в первую очередь информационных, алгоритмических и программных);
3. практическое освоение интегрированных сред изучаемого алгоритмического языка высокого уровня;
4. изучение основных этапов и принципов создания программного продукта, конструктивных компонентов и структур данных, способов их представления и обработки;
5. изучение методов обработки исключений, ошибок и подходов к отладки программных продуктов.

Задачи дисциплины:

- Изучение базовых концепций и технологий программирования, выступающих, как составная часть технологии разработки объектов профессиональной деятельности (программных продуктов);
- Получение студентами базовых знания об организации процессов управления разработкой и отладкой программных продуктов командой разработчиков и их взаимодействия с постановщиками и заказчиками программного продукта;
- Получение студентами умений и навыков использования интегрированных сред разработки, применяемых для решения практических задач создания программных продуктов.

2. Указания по проведению практических занятий

Практическая работа 1-2.

Вид практического занятия: *решение ситуационных задач.*

Образовательные технологии: *информационно-коммуникативные.*

Тема и содержание занятия: *Разработка однофайловых программ в MinGW Code::Blocks/Visual Studio Code.*

Продолжительность занятия– 4 ч.

Практическая работа 3-4.

Вид практического занятия: *решение ситуационных задач.*

Образовательные технологии: *информационно-коммуникативные*

Тема и содержание занятия: *Разработка многофайловых проектов в MinGW Code::Blocks/Visual Studio Code.*

Продолжительность занятия– 4 ч.

Практическая работа 5-6.

Вид практического занятия: *решение ситуационных задач.*

Образовательные технологии: *информационно-коммуникативные*

Тема и содержание занятия: *Разработка структурных программ в JetBrains PyCharm/Jupyter Notebook.*

Продолжительность занятия— 4 ч.

Практическая работа 7-8.

Вид практического занятия: *решение ситуационных задач.*

Образовательные технологии: *информационно-коммуникативные*

Тема и содержание занятия: *Разработка объектно-ориентированных программ в JetBrains PyCharm/Jupyter Notebook.*

Продолжительность занятия— 4 ч.

Практическая работа 9-10.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа.

Тема и содержание занятия: Определение оптимальной технологии разработки программного продукта из предложенного списка задач на языке JAVA.

Продолжительность занятия— 4 ч.

Практическая работа 11-12.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа.

Тема и содержание занятия: Разработка однофайловых программ на JAVA в среде Eclipse.

Продолжительность занятия— 4 ч.

Практическая работа 13-14.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: информационно-коммуникативные

Тема и содержание занятия: Разработка многофайловых проектов на JAVA в среде Eclipse.

Продолжительность занятия— 4 ч.

Практическая работа 15-16.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: информационно-коммуникативные

Тема и содержание занятия: Решение и разбор сложных задач на языке JAVA в среде Eclipse.

Продолжительность занятия— 4 ч.

Практическая работа 17-18.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: информационно-коммуникативные

Тема и содержание занятия: Разработка многофайловых проектов на PHP в среде NetBeans/PHPStorm.

Продолжительность занятия– 4 ч.

Практическая работа 19-20.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: информационно-коммуникативные

Тема и содержание занятия: Решение и разбор сложных задач на языке PHP в среде NetBeans/PHPStorm.

Продолжительность занятия– 4 ч.

3. Указания по проведению лабораторного практикума

Не предусмотрены.

4. Указания по проведению самостоятельной работы обучающихся

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1.	Основные технологии разработки программного обеспечения	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (разработка собственных компонентов и их установка в систему).
2.	Оценка качества процессов создания ПО. Подходы к управлению процессами разработки ПО	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (разработка собственных компонентов и их установка в систему).
3.	Среды программирования Code::Blocks и Visual Studio Code	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (разработка собственных компонентов и их установка в систему).
4.	Среды программирования JetBrains PyCharm и Jupyter Notebook	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (создание собственных библиотек).
5.	Среда программирования Eclipse	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (создание собственных библиотек).
6.	Среды программирования NetBeans и PHPStorm	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (создание собственных библиотек).

5. Указания по проведению контрольных работ для обучающихся очной формы обучения

Контрольная работа – письменная работа небольшого объема, предполагающая проверку знаний заданного к изучению материала и навыков его практического применения. Контрольные работы могут состоять из одной или нескольких практических задач с применением программных средств специального и общего назначения на компьютерах

Написание контрольной работы практикуется в учебном процессе в целях приобретения обучающимися необходимой профессиональной подготовки, развития умения и навыков самостоятельного поиска решений.

5.1. Требования к структуре

Контрольная работа имеет следующую структуру:

- титульный лист;
- оглавление;
- текст работы, структурированный по заданиям/главам (параграфам, разделам);
- заключение (при необходимости);
- библиография (список источников, при необходимости);
- приложения (при необходимости).

5.2. Требования к оформлению

Каждая контрольная работа содержит определенное количество задач. При выполнении их необходимо придерживаться следующих правил:

1. Каждую контрольную работу обучающиеся выполняют в отдельной тетради и на персональном компьютере (практические вопросы).
2. В заголовке работы должны быть разборчиво написаны: фамилия, имя и отчество, учебный шифр, номер контрольной работы (ее части), название дисциплины. Заголовок надо поместить на обложку тетради. Здесь же указать дату выполнения контрольной работы.
3. Решение задач надо располагать в порядке номеров, указанных в задании, сохраняя номер задач своего варианта.
4. Перед решением каждой задачи надо полностью выписать ее условие, заменив, где надо, общие данные контрольными из своего варианта.
5. Решения задач излагать аккуратно, объясняя основные действия, выписывая нужные формулы, делая необходимые чертежи (блок схемы).
6. После получения прорецензированной работы исправьте все ошибки и недочеты, вписав исправления на оставленных чистых страницах.

Работа засчитывается, если она при проверке (или после устранения недочетов) преподавателем получает положительную оценку (зачет). Студенты, не получившие зачета по контрольной работе, к экзамену не допускаются. Зачетные контрольные работы обязательно предъявляются на экзамене.

6. Перечень основной и дополнительной учебной литературы

Основная литература:

- 1) Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.: КУРС, НИЦ ИНФРА-М, 2017. - 240 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-906818-25-6
- 2) <http://znanium.com/catalog.php?item=bookinfo&book=766771>
- 3) Слабнов В. Д. Программирование на С++ / В. Д. Слабнов; В.Д. Слабнов. - Казань: Познание, 2012. - 136 с. - ISBN 978-5-8399-0386-9.
- 4) URL: <http://biblioclub.ru/index.php?page=book&id=364222>
- 5) Основы алгоритмизации и программирования на Python: учеб. пособие / С.Р. Гуриков. — М.: ФОРУМ: ИНФРА-М, 2017. — 343 с. — (Высшее образование: Бакалавриат).
- 6) <http://znanium.com/catalog.php?item=bookinfo&book=924699>
- 7) Карепова, Е.Д. Основы многопоточного и параллельного программирования: учебное пособие / Е.Д. Карепова; Сибирский Федеральный университет, Федеральное государственное бюджетное учреждение науки «Институт вычислительного моделирования Сибирского отделения Российской академии наук», Сибирский научно-образовательный центр суперкомпьютерных технологий. – Красноярск: СФУ, 2016. – 355 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=497217>

Дополнительная литература:

- 1) Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 400 с. — (Высшее образование: Бакалавриат). <http://znanium.com/catalog.php?item=bookinfo&book=924760>
- 2) Горелов С.В. Современные технологии программирования: разработка Windows-приложений на языке С#: учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика» (09.03.03 — для бакалавров, 09.04.03 — для магистров): в 2 т.: [16+] / С.В. Горелов; под науч. ред. П.Б. Лукьянова; Финансовый университет при Правительстве Российской Федерации. – Москва: Прометей, 2019. – Том 1. – 363 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=576037>

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

Интернет-ресурсы:

- <http://www.codeblocks.org/>
- <https://code.visualstudio.com/>
- <http://www.cplusplus.com/> - Сеть ресурсов, посвященных языку С++
- <http://dia-installer.de/>
- <https://www.jetbrains.com/> - Сайт компании-разработчика JetBrains
- <https://notepad-plus-plus.org/>
- <https://www.python.org/> – Сайт разработчиков Python
- <https://jupyter.org/>

- <https://colab.research.google.com/>
- <http://pythontutor.ru/> - Курс по программированию на языке Python
- <https://netbeans.apache.org/front/main/download/index.html>
- <https://www.netbeans.info/products/ide/features.html>
- <https://www.jetbrains.com/ru-ru/phpstorm/>
- <https://www.jetbrains.com/ru-ru/phpstorm/documentation/>
- <https://www.eclipse.org/downloads/>
- https://www.tutorialspoint.com/eclipse/eclipse_quick_guide.htm
- <https://stepik.org> Stepik — образовательная платформа и конструктор онлайн-курсов.
- <http://www.znaniium.com/> - Электронная библиотека издательства "Инфра-М"

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

Перечень программного обеспечения: Dia, Double Commander, NotePad++, Microsoft Office или свободно распространяемые аналоги, VS Code, GCC + MinGW, Python 3, PyCharm, Jupyter Notebook, Eclipse, PHPStorm, NetBeans.

Информационные справочные системы: Электронные ресурсы образовательной среды Университета.