



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ
Проректор по учебной работе
Е.К. Самаров
« 05 » 2021г.



***ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН***

**РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ
«ТЕХНОЛОГИИ И СРЕДЫ ПРОГРАММИРОВАНИЯ»**

Направление подготовки: 01.03.02.Прикладная математика и информатика

Профиль: Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавр

Форма обучения: очная




Королев
2021

Автор: Вилисов В.Я. Рабочая программа дисциплины: Технологии и среды программирования. – Королев МО: МГОТУ, 2021

Рецензент: к.т.н., доцент Бугай И.В.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки бакалавров 01.03.02 «Прикладная математика и информатика» и Учебного плана, утвержденного Ученым советом МГОТУ. Протокол № 13 от 22 июля 2021 года.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Бугай И.В. к.т.н., доцент 	<i>Бугай И.В.</i> к.т.н., доцент 	<i>Бугай И.В.</i> к.т.н., доцент 	
Год утверждения (переподтверждения)	2021	<i>2022</i>	<i>2023</i>	
Номер и дата протокола заседания кафедры	<i>№10 от 28.05.21</i>	<i>№11 от 10.06.22</i>	<i>№9 от 25.04.23</i>	

Рабочая программа согласована:

Руководитель ОПОП ВО  к.т.н., доц. И.В. Бугай

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переподтверждения)	2021	<i>2022</i>	<i>2023</i>	
Номер и дата протокола заседания УМС	<i>№7 от 15.06.21</i>	<i>№5 от 21.06.22</i>	<i>№6 от 16.05.23</i>	

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

Целью изучения дисциплины является:

1. формирование способности к восприятию, обобщению и анализу информации, необходимой для оценки вариантов, обеспечения и поддержки принятия эффективных конструкторских и управленческих решений;
2. освоение необходимого инструментария, применяемого при оценке вариантов, выборе эффективных конструкторских и управленческих решений;
3. формирование готовности применять методы математического анализа и моделирования сложных систем и процессов в профессиональной деятельности.

В процессе обучения студент приобретает и совершенствует следующие компетенции:

общефессиональные компетенции (ОПК):

- Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-4);

профессиональные компетенции (ПК):

- Способность демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий (ПК-2).

Основными **задачами** дисциплины являются:

- Освоение студентами теоретических методов дисциплины, применяемых при оценке вариантов и выборе эффективных конструкторских и управленческих решений;
- Получение студентами умений и навыков, применяемых для решения практических задач оценки вариантов и выбора эффективных конструкторских и управленческих решений.

После завершения освоения данной дисциплины студент должен:

Знать:

- особенности разработки компонентов и библиотек в системах ООП
- методы системной интеграции гетерогенных программных средств

Уметь:

- пользоваться всеми необходимыми методами разработки компонентов и библиотек в системах ООП при решении профессиональных задач;
- использовать методы системной интеграции гетерогенных программных средств при решении профессиональных задач;

Владеть:

- методами разработки компонентов и библиотек в системах ООП для успешного решения профессиональных задач и содержательной интерпретации полученных результатов
- методами системной интеграции гетерогенных программных средств для содержательной интерпретации полученных результатов

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Технологии и среды программирования» относится к дисциплинам по выбору части, формируемой участниками образовательных отношений, основной профессиональной образовательной программы подготовки бакалавров по направлению подготовки 01.03.02. «Прикладная математика и информатика».

Дисциплина базируется на ранее изученных дисциплинах: «Информатика» (школьный курс) и «Математика» (школьный курс).

Знания и компетенции, полученные при освоении дисциплины, являются базовыми для изучения дисциплин: «Математическое моделирование технических систем и процессов», «Математические методы экспертных систем», «Системы поддержки принятия решений» и др., и выполнения выпускной квалификационной работы бакалавра.

3. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины для студентов очной формы составляет 3 зачетных единицы, 108 часов.

Таблица 1

Виды занятий	Всего часов	Семестр первый	Семестр второй
ОЧНАЯ ФОРМА ОБУЧЕНИЯ			
Общая трудоемкость	108	108	
Аудиторные занятия	32	32	
Лекции (Л)	16	16	
Практические занятия (ПЗ)	16	16	
Лабораторные работы (ЛР)	-	-	
Самостоятельная работа	76	76	
Курсовые работы (проекты)	-	-	
Расчетно-графические работы	-	-	
Контрольная работа, домашнее задание	+	+	
Текущий контроль знаний	Тест	Тест	
Вид итогового контроля	Зачёт	Зачёт	
ЗАОЧНАЯ ФОРМА НЕ ПРЕДУСМОТРЕНА УЧЕБНЫМ ПЛАНОМ			

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час	Лабораторные занятия, час	Занятия в интерактивной форме, час	Код компетенций
Тема 1. Разработка приложений в IDE Code::Blocks MinGW	8	8	-	3	ОПК-4
Тема 2. Разработка в IDE JetBrains PyCharm	8	8	-	3	ПК-2
Итого:	16	16	-	6	

4.2. Содержание тем дисциплины

Тема 1. Разработка приложений в IDE MinGW Code: Blocks.

- 1.1.** Элементы структурного программирования. Конструкции языка программирования: алфавит, типы переменных, структура программы, операторы, стандартные функции и процедуры, модули.
- 1.2.** Визуальная среда для создания приложений MinGW Code: Blocks. Интерфейс. Функционал. Создание, сохранение и открытие файла программы. Запуск и отладка приложения.
- 1.3.** Программный проект. Включение файлов.
- 1.4.** Подготовка приложения к выпуску. Тестирование и отладка приложений.

2. Тема 2. Разработка в IDE JetBrains PyCharm.

- 2.1.** Разработка с использованием абстрактных типов данных.
- 2.2.** Визуальная среда для создания приложений MinGW Code: Blocks. Интерфейс. Функционал. Создание, сохранение и открытие файла программы. Запуск и отладка программы
- 2.3.** Особенности отладки и тестирования с IDE JetBrains PyCharm

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

«Методические указания для обучающихся по освоению дисциплины».

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

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Технологии и среды программирования» приведен в Приложении 1 к настоящей рабочей программе.

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. Кареева, Е.Д. Основы многопоточного и параллельного программирования: учебное пособие / Е.Д. Кареева; 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/>
- <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
- <http://pythontutor.ru/> - Курс по программированию на языке Python
- <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++, MSOffice, GCC + MinGW, Python 3, PyCharm

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 МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН*

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

(Приложение 1 к рабочей программе)

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавр

Форма обучения: очная

Королев
2021

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:		
				Знать	уметь	владеть
1.	ОПК-4	Способен решать задачи профессиональной деятельности с использованием существующих информационных коммуникационных технологий и с учетом основных требований информационной безопасности	Тема 1. Разработка приложений в IDE Code::Blocks MinGW	Классификацию и подходы к решению стандартных задач профессиональной деятельности	Пользоваться структурными методами разработки программ при решении профессиональных задач;	Инструментами разработки программ и проектов в IDE MinGW Code::Blocks для успешного решения профессиональных задач и содержательной интерпретации полученных результатов
2.	ПК-2	Способность демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий	Тема 2. Разработка в IDE JetBrains PyCharm	Классификацию и состав системного и прикладного программного обеспечения	Использовать технологии разработки, отладки и оптимизации программ и проектов	Функциями создания и отладки проектов в IDE JetBrains PyCharm для разработки прикладных и системных программ

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-4, ПК-2	Письменное задание	<p>А) полностью сформирована 5 баллов</p> <p>В) частично сформирована 3-4 балла</p> <p>С) не сформирована 2 балла</p>	<p>1. Проводится в форме практической работы, выполняемой на компьютере.</p> <p>2. Время, отведенное на процедуру – 540 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <p>1. Соответствие работы уровню формирования компетенции (0-5 баллов).</p> <p>Максимальная сумма баллов - 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не</p>

			позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.
--	--	--	---

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

Примерная тематика письменных заданий, выполняемых с применением программных средств специального и общего назначения на компьютерах:

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

Программой предусмотрены следующие виды контроля: два текущих контроля знаний в форме тестирования и итогового контроля в форме зачёта

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
Согласно графика учебного процесса	Тестирование 1, 2	ОПК-4 ПК-2	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
Согласно графика учебного процесса	Зачёт	ОПК-4 ПК-2	2 вопроса и 1 задача	Зачёт проводится в устной форме, путем ответа на вопросы. Время, отведенное на процедуру – 45 минут.	Результаты предоставляются в день проведения зачёта	Критерии оценки: «Зачтено»: 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»: демонстрирует частичные знания по темам дисциплины или незнание основных понятий; неумение использовать и применять полученные знания на практике; • не работал на практических занятиях

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 правильного ответа.

Python

1. Как создать окно?

- A. `window = newWindow()`
- B. `window = Window()`
- C. `window = Frame()`
- +D. `window = Tk()`

2. Как создать фрейм?

- A. `frame = newWindow()`
- B. `frame = Window()`
- +C. `frame = Frame()`
- D. `frame = Tk()`

3. Чтобы создать метку из родительского окна, используется _____.

- A. `label = Label(text = "Welcome to Python")`
- +B. `label = Label(window, text = "Welcome to Python")`
- C. `label = Label(text = "Welcome to Python", fg = "red")`
- D. `label = Label(text = "Welcome to Python", fg = "red", bg = "white")`

4. Чтобы поместить кнопку в указанную строку и столбец в родительский контейнер, используется _____.

- A. pack manager
- +B. grid manager
- C. place manager
- D. flow manager

5. Можно создать картинку из файла с расширением _____.

- A. `.png`
- +B. `.gif`
- C. `.bmp`
- D. `.jpg`

6. Событие _____ возникает при двойном правом щелчке мышки.

- A. `<Double-Button-1>`
- B. `<Double-Button-2>`
- +C. `<Double-Button-3>`
- D. `<Triple-Button-1>`
- E. `<Triple-Button-2>`

7. Чтобы отобразить окно диалога "Программирование на Python", используется _____.

- +A. `tkinter.messagebox.showinfo("showinfo", "Программирование на Python")`
- B. `tkinter.messagebox.showwarning("showwarning", "Программирование на Python")`
- C. `tkinter.messagebox.showerror("showerror", "Программирование на Python")`

D. `tkinter.messagebox.askyesno("ashyesno", "Программирование на Python")`

8. Для показа диалогового окна ввода "Это ПМИ-17?", используется _____

A. `tkinter.messagebox.showinfo("showinfo", "Это ПМИ-17?")`

B. `tkinter.messagebox.showwarning("showwarning", "Это ПМИ-17?")`

C. `tkinter.messagebox.showerror("showerror", "Это ПМИ-17?")`

+D. `tkinter.messagebox.askyesno("ashyesno", "Это ПМИ-17?")`

9. Функция _____.

A. должна иметь хотя бы один параметр

+B. может не иметь параметров

C. должна содержать команду `return`

10. Есть ли синтаксические ошибки при вызове функции?

```
import math
```

```
def main():
```

```
    math.sin(math.pi)
```

```
main()
```

A. Да

+B. нет

11. Есть незавершенный код:

```
def f(number):
```

```
    # Пропущенное тело функции
```

```
print(f(5))
```

Тело функции должно содержать _____.

A. `return "number"`

B. `print(number)`

C. `print("number")`

+D. `return number`

12. Что будет выведено на дисплей?

```
x = 1
```

```
def f1():
```

```
    y = x + 2
```

```
    print(y)
```

```
f1()
```

```
print(x)
```

A. 1 3

+B. 3 1

C. Возникнет ошибка времени выполнения из-за неопределенной переменной `x`.

D. 1 1

E. 3 3

13. Какие программы можно отнести к системному программному обеспечению:

+A. операционные системы;

B. прикладные программы;

C. игровые программы.

14. Какой этап выполняется раньше:

- A. отладка;
- B. оптимизация;
- +C. программирование;
- D. тестирование.

15. Самый важный критерий качества программы:

- +A. работоспособность;
- B. надежность;
- C. эффективность;
- D. быстродействие;
- E. простота эксплуатации.

16. Последовательность этапов программирования:

- +A. компилирование, компоновка, отладка;
- B. компоновка, отладка, компилирование;
- C. отладка, компилирование, компоновка;
- D. компилирование, отладка, компоновка.

17. Инструментальные средства программирования:

- +A. компиляторы, интерпретаторы;
- B. СУБД (системы управления базами данных);
- C. BIOS (базовая система ввода-вывода);
- D. ОС (операционные системы).

18. На языке программирования составляется:

- +A. исходный код;
- B. исполняемый код;
- C. объектный код;
- D. алгоритм.

19. Правила, которым должна следовать программа это:

- +A. алгоритм;
- B. структура;
- C. спецификация;
- D. состав информации.

20. Доступ, при котором записи файла читаются в физической последовательности, называется:

- +A. последовательным;
- B. прямым;
- C. простым;
- D. основным.

21. Что определяет выбор языка программирования:

- +A. область приложения;
- B. знание языка;
- C. наличие дополнительных библиотек.

22. Причины синтаксических ошибок:

- +A. плохое знание языка программирования;
- B. ошибки в исходных данных;
- C. ошибки, допущенные на более ранних этапах;

D. неправильное применение процедуры тестирования.

23. Когда можно обнаружить синтаксические ошибки:

+A. при компиляции;

B. при отладке;

C. при тестировании;

D. на этапе проектирования;

E. при эксплуатации.

24. Ошибки компоновки заключаются в том, что:

+A. указано внешнее имя, но не объявлено;

B. неправильно использовано зарезервированное слово;

C. составлено неверное выражение;

D. указан неверный тип переменной.

25. Вид ошибки с неправильным написанием служебных слов (операторов):

+A. синтаксическая;

B. семантическая;

C. логическая;

D. символьная.

26. Процедура поиска ошибки, когда известно, что она есть это:

+A. отладка;

B. тестирование;

C. компоновка;

D. транзакция;

E. трансляция.

27. Программа для просмотра значений переменных при выполнении программы:

+A. отладчик;

B. компилятор;

C. интерпретатор;

D. трассировка;

E. тестирование.

28. Когда программист может проследить последовательность выполнения команд программы:

+A. при трассировке;

B. при тестировании;

C. при компиляции;

D. при выполнении программы;

E. при компоновке.

29. Тестирование бывает:

+A. комплексное;

B. инструментальное;

C. визуальное;

D. алгоритмическое.

30. Трассировка это:

+A. проверка пошагового выполнения программы;

B. тестирование исходного кода;

- С. отладка модуля;
- Д. составление блок-схемы алгоритма.

31. Назначение отладки:

- +А. поиск причин существующих ошибок;
- В. поиск возможных ошибок;
- С. составление спецификаций;
- Д. разработка алгоритма.

32. Один из методов автоматизации программирования:

- А. структурное программирование;
- В. модульное программирование;
- +С. визуальное программирование;
- Д. объектно-ориентированное программирование.

33. Нахождение наилучшего варианта из множества возможных:

- +А. оптимизация;
- В. тестирование;
- С. автоматизация;
- Д. отладка;

34. Критерии оптимизации:

- +А. время выполнения или размер требуемой памяти;
- В. размер программы и ее эффективность;
- С. независимость модулей;
- Д. качество программы, ее надежность.

35. В чем заключается оптимизация условных выражений:

- +А. в изменении порядка следования элементов выражения;
- В. в использовании простых логических выражений;
- С. в использовании сложных логических выражений;
- Д. в использовании операций AND, OR и NOT.

36. Оптимизация циклов заключается в:

- +А. уменьшении количества повторений тела цикла;
- В. просмотре задачи с другой стороны;
- С. упрощение задачи за счет включения логических операций.

37. Сущность оптимизации циклов:

- +А. сокращение количества повторений выполнения тела цикла;
- В. сокращение тела цикла;
- С. представление циклов в виде блок-схем;
- Д. трассировка циклов;
- Е. поиск ошибок в циклах.

38. В чем сущность модульного программирования:

- +А. в разбиении программы на отдельные функционально независимые части;
- В. в разбиении программы на отдельные равные части;
- С. в разбиение программы на процедуры и функции;

39. Рекомендуемые размеры модулей:

- +А. небольшие;
- В. большие;
- С. равные;

D. фиксированной длины.

40. В чем заключается независимость модуля:

+А. в написании, отладке и тестировании независимо от остальных модулей;

В. в разработке и написании независимо от других модулей;

С. в независимости от работы основной программы.

41. При структурном программировании задача выполняется:

+А. поэтапным разбиением на более легкие задачи;

В. без участия программиста;

С. объединением отдельных модулей программы.

42. Признаки нисходящего программирования:

+А. последовательная детализация;

В. наличие оптимизации;

С. наличие тестирования;

D. автоматизация программирования.

43. В чем заключается иерархический подход в решении задачи:

+А. в последовательном разбиении задачи на более мелкие составные части;

В. в выделении основных и второстепенных элементов;

С. в возможности параллельного выполнения отдельных частей задачи.

4.2. Типовые вопросы, выносимые на зачёт

1. Понятие жизненного цикла программы и его этапы.
2. Структурное проектирование программных продуктов и его методы.
3. Теория и методы структурного программирования.
4. Методы восходящей и нисходящей разработки структуры программы.
5. Основные управляющие конструкции структурного программирования.
6. Основные понятия объектно-ориентированного программирования. Объект, свойства объекта, метод обработки, событие, класс объектов.
7. Правила хорошего стиля. Требования к стилю написания программы
8. Модульное программирование как метод разработки программ.
9. Интерфейс пользователя программного продукта. Инструментарий создания интерфейса пользователя
10. Оптимизация программы на этапе отладки. Принципы и приемы оптимизации.
11. Понятие об ошибках программного обеспечения. Источники и классификация ошибок программного обеспечения
12. Основные пути и методы борьбы с ошибками программного обеспечения. Обнаружение и локализация ошибок ввода и обработки данных.
13. Разработка процедур, предотвращающих появление ошибок. Перехват и обработка ошибок. Коды ошибок.
14. Контроль вводимых значений с помощью обработки события.
15. Процедура обработки ошибок.
16. Понятие отладки программы. Составляющие процесса отладки.
17. Отладка программы. Ошибки компиляции. Ошибки выполнения.
18. Логические ошибки.
19. Пошаговое выполнение программы.

20. Точка прерывания. Вывод значений свойств и переменных.
21. Программный способ вывода значений в окно.
22. Сущность и необходимость тестирования программного обеспечения.
23. Стадии тестирования. Виды тестовых проверок. Виды тестирования.
24. Методы структурного тестирования. Принцип «белого ящика».
25. Тестирование программ методом «черного ящика».
26. Виды программ и виды программных документов. Стадии разработки.
27. Государственный стандарт по разработке алгоритмов. Описание схем, описание символов. Правила применения символов и выполнения схем
28. Состав проекта в MinGW Code::Blocks
29. Состав проекта в JetBrains PyCharm
30. Описание работы в Idle

*ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН*

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«ТЕХНОЛОГИИ И СРЕДЫ ПРОГРАММИРОВАНИЯ»**

(Приложение 2 к рабочей программе)

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавр

Форма обучения: очная

Королев
2021

1. Общие положения

Цель дисциплины:

1. формирование способности к восприятию, обобщению и анализу информации, необходимой для оценки вариантов, обеспечения и поддержки принятия эффективных конструкторских и управленческих решений;
2. освоение необходимого инструментария, применяемого при оценке вариантов, выборе эффективных конструкторских и управленческих решений;
3. формирование готовности применять методы математического анализа и моделирования сложных систем и процессов в профессиональной деятельности.

Задачи дисциплины:

1. Освоение студентами теоретических методов дисциплины, применяемых при оценке вариантов и выборе эффективных конструкторских и управленческих решений;
2. Получение студентами умений и навыков, применяемых для решения практических задач оценки вариантов и выбора эффективных конструкторских и управленческих решений.

2. Указания по проведению практических занятий

Практическая работа 1.

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

Продолжительность занятия– 4 ч.

Практическая работа 2.

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

Продолжительность занятия– 4 ч.

Практическая работа 3.

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

Продолжительность занятия– 4 ч.

Практическая работа 4.

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

Продолжительность занятия– 4 ч.

3. Указания по проведению лабораторного практикума

Не предусмотрено учебным планом.

4. Указания по проведению самостоятельной работы студентов

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1.	Разработка приложений в IDE Code::Blocks MinGW	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.	Разработка в IDE JetBrains PyCharm	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 студентов факультета заочного обучения

Учебным планом не предусмотрено.

6. Указания по проведению курсовых работ

Не предусмотрено учебным планом.

7. Перечень основной и дополнительной учебной литературы

Основная литература:

5. 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>

6. Слабнов В. Д. Программирование на C++ / 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>

7. 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>

8. Кареева, Е.Д. Основы многопоточного и параллельного программирования: учебное пособие / Е.Д. Кареева; 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:

3. 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>

4. 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:

1. <http://www.codeblocks.org/>
2. <http://www.cplusplus.com/> - Сеть ресурсов, посвященных языку С++
3. <http://dia-installer.de/>
4. <https://www.jetbrains.com/> - Сайт компании-разработчика JetBrains
5. <https://notepad-plus-plus.org/>
6. <https://www.python.org/> – Сайт разработчиков Python
7. <http://pythontutor.ru/> - Курс по программированию на языке Python
8. <https://stepik.org> Stepik — образовательная платформа и конструктор онлайн-курсов.
9. <http://www.znanium.com/> - Электронная библиотека издательства "Инфра-М"

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

Перечень программного обеспечения: Dia, Double Commander, NotePad++, MSOffice, GCC + MinGW, Python 3, PyCharm

Информационные справочные системы: Электронные ресурсы образовательной среды Университета.