



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: 01.03.02. Прикладная математика и информатика

Профиль: Программирование, математическое моделирование

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

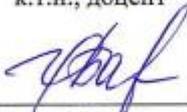
Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

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

Рецензент: к.т.н., доцент Баранова О.М.

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 Университета. Протокол № 9 от 11.04.2023 г.

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			
Номер и дата протокола заседания кафедры	№7 от 03.04.2023			

Рабочая программа согласована:

Руководитель ОПОП ВО _____



И.В. Бугай, к.т.н., доцент

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2023			
Номер и дата протокола заседания УМС	№5 от 11.04.2023			

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:

профессиональные компетенции (ПК):

- Способность демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий (ПК-2);
- Способность учитывать знания проблем и тенденций развития рынка ПО в профессиональной деятельности (ПК-4);
- Способность создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий, программирования и компьютерной техники (ПК-5).

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.

Показатель освоения компетенции отражают следующие индикаторы:

Необходимые знания:

- Знать возможности существующей программно-технической архитектуры
- Знать возможности современных и перспективных средств разработки программных продуктов, технических средств
- Знать методологии разработки программного обеспечения и технологии программирования
- Знать методологии и технологии проектирования и использования баз данных
- Знать основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных

продуктов и программных комплексов, их сопровождения, администрирования и развития

- Обладать базовыми знаниями, полученными в области математических или естественных наук, программирования или информационных технологий

Необходимые умения:

- Уметь проводить анализ исполнения требований
- Уметь вырабатывать варианты реализации требований
- Уметь использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта
- Уметь находить, формулировать и решать стандартные задачи в научно-исследовательской деятельности в математике и информатике;

Трудовые действия:

- Проводить оценку и обоснование рекомендуемых решений
- Осуществлять коммуникации с заинтересованными сторонами
- Иметь практический опыт применения указанных выше методов и технологий
- Иметь практический опыт научно-исследовательской деятельности в математике и информатике.

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».

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

Знания и компетенции, полученные при освоении дисциплины, являются базовыми для изучения дисциплин: «Операционные системы, среды и оболочки», «Программирование микропроцессоров», «Математические методы и среды разработки экспертных систем», «Технологии и системы коллективной разработки программ» 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. Практическая подготовка обучающихся составляет 16 часов.

Таблица 1

Виды занятий	Всего часов	Семестр первый
Общая трудоемкость	108	108
Аудиторные занятия	48	48
Лекции (Л)	16	16
Практические занятия (ПЗ)	16	16

Лабораторные работы (ЛР)	16	16
Практическая подготовка	16	16
Самостоятельная работа	60	60
Курсовые работы (проекты)	-	-
Расчетно-графические работы	-	-
Контрольная работа	+	+
Текущий контроль знаний	Тест	Тест
Вид итогового контроля	Экзамен	Экзамен
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 работы	Занятия в интерактивной форме, час	Практическая подготовка	Код компетенций
Тема 1. Основные технологии разработки программного обеспечения	4	2	4	2	4	ПК-4
Тема 2. Оценка качества процессов создания ПО. Подходы к управлению процессами разработки ПО	4	2	4	2	4	ПК-5
Тема 3. Среды программирования Code::Blocks и Visual Studio Code	4	6	4	2	4	ПК-2
Тема 4. Среды программирования JetBrains PyCharm и Jupyter Notebook	4	6	4	2	4	ПК-2
Итого:	16	16	16	8	16	

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.

2. 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.

3. Тема 3. Среда 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.

4. Тема 4. Среда 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

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 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>

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 C#: 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 C++
- <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://stepik.org> Stepik — образовательная платформа и конструктор онлайн-курсов.
- <http://www.znanium.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, VS Code, GCC + MinGW, Python 3, PyCharm, Jupyter Notebook.

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: 01.03.02 Прикладная математика и информатика

Профиль: Программирование, математическое моделирование

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 приобретает:		
				Необходимые знания	Необходимые умения	Трудовые действия
1.	ПК-4	Способность учитывать знания проблем и тенденций развития рынка ПО в профессиональной деятельности	Тема 1. Основные технологии разработки программного обеспечения	Знать возможности существующей программно-технической архитектуры Знать возможности современных и перспективных средств разработки программных продуктов, технических средств Знать методологии разработки программного обеспечения и технологии программирования Знать методологии и технологии проектирования и использования баз данных	Уметь проводить анализ исполнения требований Уметь выработать варианты реализации требований	Проводить оценку и обоснование рекомендуемых решений Осуществлять коммуникации с заинтересованными сторонами
2.	ПК-5	Способность создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий, программирования и компьютерной техники	Тема 2. Оценка качества процессов создания ПО. Подходы к управлению процессами разработки ПО	Знать основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных	Уметь использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта	Иметь практический опыт применения указанных выше методов и технологий

				комплексов, их сопровождения, администрирования и развития		
3.	ПК-2	Способность демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий	Тема 3. Среды программирования Code::Blocks и Visual Studio Code Тема 4. Среды программирования JetBrains PyCharm и Jupyter Notebook	Обладать базовыми знаниями, полученными в области математических или естественных наук, программирования или информационных технологий	Уметь находить, формулировать и решать стандартные задачи в научно-исследовательской деятельности в математике и информатике	Иметь практический опыт научно-исследовательской деятельности в математике и информатике

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
ПК-2, ПК-4, ПК-5	Тест	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 90% правильных ответов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 70% правильных ответов; •компетенция освоена на базовом уровне – от 51% правильных ответов; <p>В) не сформирована (компетенция не освоена) – менее 50% правильных ответов</p>	<p>Проводится письменно</p> <p>Время, отведенное на процедуру –30 мин.</p> <p>Неявка 0 баллов.</p> <p>Критерии оценки определяются процентным соотношением.</p> <p>Неудовлетворительно – менее 50% правильных ответов.</p> <p>Удовлетворительно – от 51% правильных ответов.</p> <p>Хорошо – от 70%.</p> <p>Отлично – от 90%.</p> <p>Максимальная оценка – 5 баллов.</p>
	Выполнение контрольной работы	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>При определении сформированности компетенций критериями оценивания выступают методические рекомендации, разработанные по дисциплине для данного вида.</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. 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

Е. 3 3

13. Какие программы можно отнести к системному программному обеспечению:

- +А. операционные системы;
- В. прикладные программы;
- С. игровые программы.

14. Какой этап выполняется раньше:

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

15. Самый важный критерий качества программы:

- +А. работоспособность;
- В. надежность;
- С. эффективность;
- Д. быстродействие;
- Е. простота эксплуатации.

16. Последовательность этапов программирования:

- +А. компилирование, компоновка, отладка;
- В компоновка, отладка, компилирование;
- С. отладка, компилирование, компоновка;
- Д. компилирование, отладка, компоновка.

17. Инструментальные средства программирования:

- +А. компиляторы, интерпретаторы;
- В.СУБД (системы управления базами данных);
- С. BIOS (базовая система ввода-вывода);
- Д. ОС (операционные системы).

18. На языке программирования составляется:

- +А. исходный код;
- В. исполняемый код;
- С. объектный код;
- Д. алгоритм.

19. Правила, которым должна следовать программа это:

- +А. алгоритм;
- В. структура;
- С. спецификация;
- Д. состав информации.

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

- +А. последовательным;
- В. прямым;
- С. простым;
- Д. основным.

21. Что определяет выбор языка программирования:

- +А. область приложения;

В. знание языка;

С. наличие дополнительных библиотек.

22. Причины синтаксических ошибок:

+А. плохое знание языка программирования;

В. ошибки в исходных данных;

С. ошибки, допущенные на более ранних этапах;

Д. неправильное применение процедуры тестирования.

23. Когда можно обнаружить синтаксические ошибки:

+А. при компиляции;

В. при отладке;

С. при тестировании;

Д. на этапе проектирования;

Е. при эксплуатации.

24. Ошибки компоновки заключаются в том, что:

+А. указано внешнее имя, но не объявлено;

В. неправильно использовано зарезервированное слово;

С. составлено неверное выражение;

Д. указан неверный тип переменной.

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

+А. синтаксическая;

В. семантическая;

С. логическая;

Д. символьная.

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

+А. отладка;

В. тестирование;

С. компоновка;

Д. транзакция;

Е. трансляция.

27. Программа для просмотра значений переменных при выполнении программы:

+А. отладчик;

В. компилятор;

С. интерпретатор;

Д. трассировка;

Е. тестирование.

28. Когда программист может проследить последовательность выполнения команд программы:

+А. при трассировке;

В. при тестировании;

С. при компиляции;

Д. при выполнении программы;

Е. при компоновке.

29. Тестирование бывает:

+А. комплексное;

- В. инструментальное;
- С. визуальное;
- Д. алгоритмическое.

30. Трассировка это:

- +А. проверка пошагового выполнения программы;
- В. 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;

Д. 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;

Д. 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.

3.2 Примерная тематика контрольных 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 3х линейных уравнений с 3мя 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 АД "Стек". Использовать модуль.

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

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

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

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

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

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

23. Разработать приложение для демонстрации работы с АД "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 экзамена

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
	Тестирование 1, 2	ПК-2 ПК-4 ПК-5	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
Проводится в сроки, установленные графиком образовательного процесса	Экзамен	ПК-2 ПК-4 ПК-5	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	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

						<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 «Удовлетворительно»: <p>демонстрирует частичные знания по темам дисциплин; незнание неумение использовать и применять полученные знания на практике; не работал на практических занятиях;</p> <p>Неудовлетворительно»:</p> <p>демонстрирует частичные знания по темам дисциплин; незнание основных понятий предмета; неумение использовать и применять полученные знания на практике; не работал на практических занятиях;</p> <ul style="list-style-type: none"> • не отвечает на вопросы.
--	--	--	--	--	--	---

4.2. Типовые вопросы, выносимые на экзамен

- 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?
- 16) Охарактеризуйте проблемы разработки сложных программных систем?
- 17) Какие специфические особенности разработки программных систем вы знаете?
- 18) В чем заключается сущность блочно-иерархического подхода к созданию системы?
- 19) Что такое декомпозиция системы?
- 20) Сущность метода пошаговой детализации?
- 21) Что понимается под локальной оптимизацией?
- 22) Что лежит в основе блочно-иерархического подхода к созданию программных систем?
- 23) Какие существуют подходы к организации создания и использованию программных систем?
- 24) Перечислите этапы процесса разработки программных систем?
- 25) Какие этапы разработки программных систем регламентируются ГОСТ19.102–77?
- 26) Какие модели жизненного цикла программного обеспечения существуют?
- 27) Что понимается под каскадной моделью жизненного цикла ПО?
- 28) Что понимается под моделью жизненного цикла ПО с промежуточным контролем?
- 29) Что понимается под спиральной моделью жизненного цикла ПО?
- 30) Что понимается под оценкой качества процессов создания ПО?
- 31) Какие группы и показатели качества ПО вы знаете?
- 32) Какие качественные характеристики ПО необходимо обеспечивать (ГОСТ Р ИСО 9126)?
- 33) Что понимается под технологичностью ПО и какие концепции положены в её основы?
- 34) Чем определяется технологичность ПО?
- 35) Что понимается под процедурной декомпозицией разрабатываемого ПО?
- 36) Что понимается под объектной декомпозицией разрабатываемого ПО?
- 37) Что такое модуль и какие требования к нему предъявляются?
- 38) Какие способы обеспечения высокого уровня технологичности разработки ПО вы знаете?

- 39) Что такое сцепление модулей, какие типы сцепления модулей существуют?
- 40) Что такое связность модулей, какие виды связности модулей существуют?
- 41) Какие библиотеки ресурсов существуют?
- 42) Что понимается под нисходящей и восходящей разработкой ПО?
- 43) Что такое Agile?
- 44) Четыре ценности Agile?
- 45) Дайте характеристики принципам Agile?
- 46) Что такое Scrum?
- 47) Сущность принципов методологии Scrum?
- 48) На чём основан Scrum. В чём отличие Scrum от Agile?
- 49) Основная терминология Scrum?
- 50) Что такое спринт в Scrum?
- 51) Содержание цикла работы Scrum?
- 52) Каким образом определяется приоритетность работ в Scrum?
- 53) Как оценить успешность внедрения методологии Agile?
- 54) Что такое Jira, для чего она используется и какие задачи решаются с её помощью?
- 55) Среда программирования Code::Blocks?
- 56) Среда программирования PyCharm?
- 57) Среда программирования Visual Studio Code?
- 58) Среда программирования Jupyter Notebook?

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: 01.03.02 Прикладная математика и информатика

Профиль: Программирование, математическое моделирование

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

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 1-2.

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.

3. Указания по проведению лабораторного практикума

Лабораторная работа 1-2.

Вид занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа.

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

Продолжительность занятия– 4 ч.

Лабораторная работа 3-4.

Вид занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа.

Тема и содержание занятия: Разработка бэглота требований к предложенному списку программных продуктов из списка задач. Моделирование Scrum-процесса. Формирование планирование спринта. Ретроспектива: анализ спринта.

Продолжительность занятия– 4 ч.

Лабораторная работа 5-6.

Вид занятия: смешанная форма практического занятия.

Образовательные технологии: информационно-коммуникативные

Тема и содержание занятия: Решение и разбор сложных задач на языке C/C++ в среде программирования Code::Blocks и Visual Studio Code.

Продолжительность занятия– 4 ч.

Лабораторная работа 7-8.

Вид занятия: смешанная форма практического занятия.

Образовательные технологии: информационно-коммуникативные

Тема и содержание занятия: Решение и разбор сложных задач на языке Python в среде программирования JetBrains PyCharm/Jupyter Notebook.

Продолжительность занятия– 4 ч.

4. Указания по проведению самостоятельной работы обучающихся

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1.	Основные технологии разработки программного обеспечения	<ol style="list-style-type: none">1. Подготовка к практическим занятиям по материалам лекций и учебной литературы.2. Выполнение практических заданий3. 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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:

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 C#: 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:

1. <http://www.codeblocks.org/>
2. <https://code.visualstudio.com/>
3. <http://www.cplusplus.com/> - Сеть ресурсов, посвященных языку C++
4. <http://dia-installer.de/>
5. <https://www.jetbrains.com/> - Сайт компании-разработчика JetBrains
6. <https://notepad-plus-plus.org/>
7. <https://www.python.org/> – Сайт разработчиков Python
8. <https://jupyter.org/>
9. <https://colab.research.google.com/>
10. <http://pythontutor.ru/> - Курс по программированию на языке Python
11. <https://stepik.org> Stepik — образовательная платформа и конструктор онлайн-курсов.
12. <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++, MSOffice, VS Code, GCC + MinGW, Python 3, PyCharm, Jupyter Notebook.

Информационные справочные системы: Электронные ресурсы образовательной среды Университета.