



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храшкин

26.01.2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММИРОВАНИЕ

Направление подготовки (специальность)

09.03.03 Прикладная информатика

Направленность (профиль/специализация) программы

Разработка компьютерных игр и AR/VR-приложений (виртуальной/дополненной
реальности)

Уровень высшего образования - бакалавриат

Форма обучения

очная

| | |
|---------------------|---|
| Институт/ факультет | Институт энергетики и автоматизированных систем |
| Кафедра | Бизнес-информатики и информационных технологий |
| Курс | 1 |
| Семестр | 1, 2 |

Магнитогорск
2022 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Рабочая программа рассмотрена и одобрена на заседании кафедры Бизнес-информатики и информационных технологий

25.01.2022, протокол № 6

Зав. кафедрой  Г.Н. Чусавитина

Рабочая программа одобрена методической комиссией ИЭиАС

26.01.2022 г. протокол № 5


Председатель  В.Р. Храмшин

Рабочая программа составлена:

доцент кафедры БИиИТ, канд. пед. наук  И.В. Гаврилова

Рецензент:

руководитель группы анализа ИТ-проектов

ЗАО "КОНСОМ СКС" , канд. техн. наук  В.А. Ошурков

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины «Программирование» является освоение студентами методики постановки, подготовки и решения инженерно-технических задач на современных вычислительных машинах с использованием различных средств программирования.

Для достижения поставленной цели в курсе «Программирование» решаются задачи приобретения:

представлений об алгоритме и конечном наборе базовых управляющих структур;
представлений о структуре программных средств;
знаний основ кодирования на языках высокого уровня;
знаний моделей визуального и событийно-управляемого программирования;
представлений об основных понятиях объектно-ориентированного подхода;
знаний современных технических и программных средств взаимодействия пользователей с компьютером, возможностей компьютера как средства исследования, автоматизации обработки данных и решения проектных и научно-технических задач.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Программирование входит в обязательную часть учебного плана образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Необходимы компетенции в области математики, физики, информатики и ИКТ, сформированные на предыдущей ступени образования (среднее профессиональное образование, среднее общее образование).

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Проектная деятельность
Подготовка к сдаче и сдача государственного экзамена
Интернет технологии
Учебная - ознакомительная практика
Вычислительные системы, сети и телекоммуникации
Учебная - технологическая (проектно-технологическая) практика
Практикум по программной инженерии
Разработка приложений на платформе 1С

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Программирование» обучающийся должен обладать следующими компетенциями:

| Код индикатора | Индикатор достижения компетенции |
|----------------|--|
| 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-2.1 | Применяет современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности |
| ОПК-7 | Способен разрабатывать алгоритмы и программы, пригодные для практического применения; |
| ОПК-7.1 | Определяет средства разработки программных средств для решения практических задач профессиональной деятельности |

| | |
|---------|--|
| ОПК-7.2 | Разрабатывает алгоритмы и программы для решения прикладных задач различных классов |
|---------|--|

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 9 зачетных единиц 324 академических часов, в том числе:

- контактная работа – 197,9 академических часов;
- аудиторная – 192 академических часов;
- внеаудиторная – 5,9 академических часов;
- самостоятельная работа – 90,4 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - зачет, экзамен

| Раздел/ тема дисциплины | Семестр | Аудиторная контактная работа (в академических часах) | | | Самостоятельная работа студента | Вид самостоятельной работы | Форма текущего контроля успеваемости и промежуточной аттестации | Код компетенции |
|---|---------|--|-----------|-------------|---------------------------------|--|---|---------------------------|
| | | Лек. | лаб. зан. | практ. зан. | | | | |
| 1. Основные конструкции языка высокого уровня | | | | | | | | |
| 1.1 История возникновения языков C и C++. Алфавит языка, константы, идентификаторы, ключевые слова. Понятие переменной, описание переменных в программе | 1 | 2 | 2/2И | | 5 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 1.2 Типы данных. Целые типы данных, классификация на знаковые и беззнаковые, представление в памяти компьютера. Вещественные типы, понятие мантиссы и порядка. Указатели, операции взятия адреса и разадресации. Перечисляемый тип данных. Тип данных void. Выражения. Операнды и операции. Унарные, бинарные и тернарные операции. Преобразования типов при вычислении выражений | | 4 | 4/1И | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |

| | | | | | | | | |
|--|---|---|-------|--|----|--|---|---------------------------------|
| 1.3 Структура С-программы. Операторы языка С. Оператор-выражение, составной оператор. Разветвляющиеся алгоритмические конструкции. Операторы if и switch. Циклические алгоритмические конструкции. Циклы с предпроверкой, с постпроверкой, с параметром, с выходом из середины. Операторы while, do while, for, break, continue. | | 2 | 8/2И | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| Итого по разделу | | 8 | 14/5И | | 13 | | | |
| 2. Основные структуры данных | | | | | | | | |
| 2.1 Массивы. Объявление в программе. Вычисление индексных выражений. Стандартные алгоритмы обработки массивов — ввод/вывод, обнуление, поиск заданного и экстремального значения, сортировка. | | 4 | 8/2И | | 5 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 2.2 Структуры. Описание в программе. Объединения (union). Битовые структуры. Вариантные структуры | | 2 | 2/1И | | 5 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 2.3 Методы доступа к элементам массивов. Указатели на многомерные массивы. Адресная арифметика. Динамическое выделение памяти. Выделение памяти в соответствие с типом указателя. Выделение памяти под нетипизированный указатель. Основные ошибки, связанные с динамическим выделением памяти. Динамические массивы. Реализация структуры — односвязный линейный список | 1 | 2 | 6 | | 5 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| Итого по разделу | | 8 | 16/3И | | 15 | | | |
| 3. Методы структурного программирования | | | | | | | | |

| | | | | | | | | |
|--|---|----|-------|--|------|--|--|---------------------------|
| 3.1 Понятие функции. Определение и прототип функции. Передача параметров | 1 | 2 | 4/2И | | 3 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 3.2 Указатели на функцию. Предварительная инициализация параметров функции. Функции с переменным числом параметров. Передача параметров функции main | | 4 | 2/2И | | 3 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| Итого по разделу | | 6 | 6/4И | | 6 | | | |
| 4. Методы объектно-ориентированного программирования | | | | | | | | |
| 4.1 Объектно-ориентированный подход к программированию. Понятие инкапсуляции. Объявление класса в языке С++. Члены класса. Ссылка this. Конструктор. Конструктор копирования. Деструктор. Создание класса «битовое множество». Доступ к членам класса. Функции — друзья класса | 1 | 4 | 6/2И | | 6 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 4.2 Наследование. Базовый и производные классы. Спецификация производных классов. Доступ к наследованным компонентам базового класса. Соотношение между базовыми и производными классами. Последовательность вызова конструкторов и деструкторов для объектов производных классов | | 4 | 6/2И | | 6 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 4.3 Перегрузка операций. Использование методов класса и дружественных функций. Перегрузка операций преобразования типов. Переопределение ввода-вывода на языке С++. Стандартный ввод/вывод. Форматируемый вывод. Ввод/вывод с диска. Ввод/вывод для типов данных, определенных пользователем | | 6 | 6/2И | | 6,1 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| Итого по разделу | | 14 | 18/6И | | 18,1 | | | |

| | | | | | | | |
|---|----|--------|---|------|--|---|---------------------------------|
| Итого за семестр | 36 | 54/18И | | 52,1 | | зачёт | |
| 5. Разработка модульного программного обеспечения | | | | | | | |
| 5.1 Современные инструментальные средства разработки модульного программного обеспечения | 4 | 6 | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 5.2 Понятие многофайлового проекта. Время жизни и область видимости локальных и глобальных объектов | 4 | 2 | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 5.3 Использование пространств имен. Стандартное пространство имен. Пользовательские пространства имен. Вложенные пространства имен | 4 | 2 | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 5.4 Директивы препроцессора. Директива #include. Директива #define. Условные директивы препроцессора | 4 | 2 | 2 | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 5.5 Шаблоны функций. Основные понятия. Параметры шаблонов. Шаблоны классов. Пример контейнера stack | 4 | 4/4И | | 6 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 5.6 Библиотека STL. Итераторы, алгоритмы, контейнеры, функциональные объекты. Контейнерные классы в со-временных компиляторах. Обработка исключительных ситуаций. Генерация исключений. Перехват исключений | 4 | 6 | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию | Проверка индивидуальных заданий тестирование | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| Итого по разделу | 24 | 22/4И | | 26 | | | |
| 6. Разработка приложений в среде программирования Qt | | | | | | | |

| | | | | | | | | |
|--|---|----|----------|--|------|---|---|---------------------------|
| 6.1 Введение в среду разработки Qt5. Стандартные библиотеки | 2 | 2 | 8 | | 2 | Изучение руководства по работе в среде Qt 5 | Проверка отчёта о выполнении лабораторной работы | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 6.2 Разработка графического пользовательского интерфейса приложения | | 2 | 12/2И | | 2 | Изучение руководства по работе в среде Qt 5 | Проверка отчёта о выполнении лабораторной работы | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 6.3 Обработка событий, работа с таймером, графикой и звуком | | 2 | 8 | | 4,3 | Работа над индивидуальным проектом | Проверка проектной документации | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 6.4 Организация доступа к базам данных | | 2 | 8/2И | | 2 | Работа над индивидуальным проектом | Проверка отчёта о работе над индивидуальным проектом | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| 6.5 Настройки приложения, организация тестирования, интернализация и миграция на различные платформы | | 2 | 10 | | 2 | Работа над индивидуальным проектом | Оценка презентации результатов работы над индивидуальным проектом | ОПК-7.1, ОПК-7.2, ОПК-2.1 |
| Итого по разделу | | 10 | 46/4И | | 12,3 | | | |
| Итого за семестр | | 34 | 68/8И | | 38,3 | | экзамен | |
| Итого по дисциплине | | 70 | 122/26 И | | 90,4 | | зачет, экзамен | |

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Проблемная лекция – изложение материала, предполагающее постановку проблемных и дискуссионных вопросов, освещение различных научных подходов, авторские комментарии, связанные с различными моделями интерпретации изучаемого материала, а также работа над индивидуальным проектом.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично-значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция-провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-прессконференция.

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2022. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492984> (дата обращения: 30.06.2022).

2. Торшина, О. А. Объектно-ориентированное программирование [Электронный ресурс] : учебное пособие / О. А. Торшина ; МГТУ. - Магнитогорск : МГТУ, 2018. - 1 электрон. опт. диск (CD-ROM). - Режим доступа: <https://magtu.informsystema.ru/uploader/fileUpload?name=3613.pdf&show=dcatalogues/1/1524595/3613.pdf&view=true>. - Макрообъект. - ISBN 978-5-9967-1132-1.

б) Дополнительная литература:

1. Практикум по программированию на языке C++ [Электронный ресурс] : учебное пособие / В. Е. Торчинский, А. Н. Калитаев, В. Д. Тутарова, Ю. В. Федосеева ; МГТУ. - Магнитогорск : МГТУ, 2017. - 1 электрон. опт. диск (CD-ROM). - Режим доступа: <https://magtu.informsystema.ru/uploader/fileUpload?name=3004.pdf&show=dcatalogues/1/1134950/3004.pdf&view=true> - Макрообъект.

2. Кочержинская, Ю. В. Технология разработки программного обеспечения [Электронный ресурс] : практикум / Ю. В. Кочержинская ; МГТУ. - Магнитогорск : МГТУ, 2017. - 1 электрон. опт. диск (CD-ROM). - Режим доступа: <https://magtu.informsystema.ru/uploader/fileUpload?name=3239.pdf&show=dcatalogues/1/1136961/3239.pdf&view=true>. - Макрообъект.

3. Программирование. Сборник задач : учебное пособие для вузов / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.] ; Под редакцией М. М. Марана. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 140 с. — ISBN 978-5-507-44322-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/223418> (дата обращения: 25.04.2022). — Режим доступа: для авториз. пользователей.

в) Методические указания:

1. Гаврилова И.В. Программирование: методические рекомендации по организации самостоятельной работы направления Прикладная информатика. Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2021. 8 с.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

| Наименование ПО | № договора | Срок действия лицензии |
|------------------------------|------------------------------|------------------------|
| 7Zip | свободно распространяемое | бессрочно |
| Git | свободно распространяемое | бессрочно |
| Eclipse | свободно распространяемое | бессрочно |
| FAR Manager | свободно распространяемое | бессрочно |
| Linux Calculate | свободно распространяемое | бессрочно |
| Calculate Linux Desktop Xfce | свободно распространяемое ПО | бессрочно |

Профессиональные базы данных и информационные справочные системы

| Название курса | Ссылка |
|--|--|
| Электронная база периодических изданий East View Information Services, ООО «ИВИС» | https://dlib.eastview.com/ |
| Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ) | URL: https://elibrary.ru/project_risc.asp |
| Поисковая система Академия Google (Google Scholar) | URL: https://scholar.google.ru/ |
| Информационная система - Единое окно доступа к информационным ресурсам | URL: http://window.edu.ru/ |

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Лекционная аудитория — мультимедийные средства хранения, передачи и представления информации.

Учебные аудитории для проведения лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации: персональные компьютеры с выходом в Интернет и с доступом в электронную информационно-образовательную среду университета. Комплекс лабораторных работ, тестовых заданий для проведения промежуточных и рубежных контролей.

Помещения для самостоятельной работы обучающихся: персональные компьютеры с выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

Помещение для хранения и профилактического обслуживания учебного оборудования: стеллажи для хранения учебно-наглядных пособий и учебно-методической документации.

Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Программирование» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает решение контрольных задач на лабораторных занятиях.

Примерные аудиторные контрольные работы (АКР):

Раздел 1.

1. По известному радиусу определить длину окружности, площадь круга, площадь поверхности сферы и объем шара.

2. Дано натуральное K — количество секунд. Определить сколько это составляет часов, минут и секунд. Например, 4000 секунд — это 1 час, 6 минут и 40 секунд. Использовать не более 4-х арифметических операций.

3. Определить сколько лет понадобится шаху, чтобы собрать урожай зерна, требуемый изобретателем шахмат. Считать, что среднегодовой сбор составляет 70 млн. тонн, а на 1 грамм приходится 10 зерен.

4. Вычислить значение выражения:

$$y = \sqrt{\frac{a \sin^2 b + \cos b^3 + \sqrt[3]{b^2}}{\sqrt[4]{\left| \frac{a \cdot \operatorname{tg} b}{1 - e^{\sqrt{a}}} \right|}}}$$

Контрольный пример: при $a=b=0.5$ $y=1.214$

5. Дано время — два целых числа количество часов и минут. Необходимо определить меньший угол между часовой и минутной стрелками на циферблате часов.

6. Дано действительное число E ($E>0$). Необходимо вычислить следующую сумму: каждое слагаемое рассчитывается по формуле: $|1/3|^n * \cos 3^{n-1}|$, где n меняется от 1 до бесконечности с шагом 1. Следует учесть только те слагаемые, в которых множитель $|1/3|^n$ имеет величину не меньшую, чем E .

7. Дано целое число $M>1$. Получить наибольшее целое k , при котором $4^k < M$.

8. Даны положительные действительные числа A, X, E . В последовательности $y(1), y(2), \dots$, образованной по закону $y(0)=A; y(i)=0.5 [y(i-1)+x/y(i-1)]$, $i=1,2,3, \dots$, найти первый член $y(n)$, для которого выполнено неравенство $\left| y(n)^2 - y(n-1)^2 \right| < E$.

9. Число Π вычисляется по формуле Грегори следующим образом: $\Pi=4(1-1/3+1/5-1/7+1/9-\dots)$, причем, чем больше слагаемых в скобках, тем выше точность

вычисления числа π . Определить минимальное количество слагаемых для вычисления π с точностью 0.01

10. Дана последовательность, состоящая из дробей: $1/1, 4/2, 7/3, 10/4, \dots$. Какое минимальное количество элементов последовательности нужно сложить, чтобы сумма превысила заданное число $S > 1$?

Раздел 2.

1. Дано N действительных случайных чисел в диапазоне от -100 до 100. Найти минимальное положительное число и максимальное отрицательное число.

2. Дано N целых случайных чисел в диапазоне от 1 до 100. Найти сумму четных и количество нечетных чисел.

3. Дано N действительных случайных чисел в диапазоне от 1 до 10. Все элементы последовательности, значение которых меньше двух, заменить на ноль, кроме того, получить сумму элементов, находящихся в диапазоне от 3 до 6, а также подсчитать их количество.

4. Дан одномерный массив из N действительных случайных чисел в диапазоне от 1 до 50. Найти минимальный элемент среди элементов с нечетным индексом и максимальный среди элементов с четным.

5. Дан одномерный массив из N случайных действительных чисел в диапазоне от -4 до 8. Вывести в порядке невозрастания (убывания) элементы, модуль которых больше 2.

6. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Отобрать кандидатов на олимпиады (с отличными оценками) по каждому из предметов.

7. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Найти среднюю в группе оценку по каждому из предметов.

8. Спортсмены на соревнованиях совершают 6 попыток в прыжках в длину. Определить лучший результат для каждого участника.

9. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Найти в группе среднюю оценку для каждого учащегося.

10. Для группы фирм известен курс их акций за каждый из месяцев календарного года. Составить список тех фирм, курс акций которых все время повышался (т.е. курс за каждый последующий месяц больше, чем за предыдущий).

Раздел 3.

1. Дано N десятков целых чисел. Определить, сколько из них могут составлять геометрическую прогрессию. Проверку оформить в виде функции.

2. Дано N десятков целых чисел. Определить, сколько из них могут составлять арифметическую прогрессию. Проверку оформить в виде функции.

3. Дано N десятков целых чисел. Определить, сколько из них могут составлять ряд Фибоначчи. Первое число Фибоначчи равно 0, второе – 1. Каждое последующее равно сумме двух предыдущих. Проверку оформить в виде функции.

4. Дано N пар чисел, представляющих собой координаты точек на плоскости. Найти R — радиус наименьшей окружности с центром в начале координат, в которую попадают все точки. Определение расстояния от точки до начала координат оформить в виде функции.

5. Известны оценки группы студентов за сессию. В группе 20 студентов, в сессии 4 экзамена. Определить суммарную стипендию. Считать, что стипендия в размере R рублей начисляется студентам, сдавшим сессию без троек, а отличники получают стипендию, повышенную на 25%. Подсчет стипендии студента оформить в виде функции.

6. Известен расход электроэнергии по всем квартирам 24-х квартирному дому. Определить суммарную плату за электричество. При расходе до 100 кВт*ч на человека берется тариф R копеек за 1 кВт*ч, в случае превышения нормы тариф возрастает на 20%. Подсчет платы для квартиры оформить в виде функции.

7. Известна ежемесячная заработная плата персонала предприятия в течение календарного года. Вывести фамилии тех сотрудников, у которых годовая заработная плата выше средней. Считать, что штат предприятия составляет 7 человек. Подсчет годовой зарплаты работника оформить в виде функции.

8. Известна ежемесячная заработная плата персонала предприятия в течение календарного года. Вывести фамилии сотрудников с минимальной и максимальной годовой заработной платой. Считать, что штат предприятия составляет 8 человек. Подсчет годовой зарплаты работника оформить в виде функции.

9. Дан одномерный массив из 100 случайных целых чисел в диапазоне от 5 до 25 включительно. Вывести все числа, которые максимально часто встречаются в массиве и количество их повторений. Подсчет количества повторений для числа оформить в виде функции.

10. Дан одномерный массив из 150 случайных целых чисел в диапазоне от 14 до 37 включительно. Вывести те числа, которые наиболее редко встречаются в массиве и количество их повторений. Подсчет количества повторений для числа оформить в виде функции.

Раздел 4.

1. Создать класс для хранения календарных дат. Обеспечить возможность работы с датами в различных форматах, изменения даты на заданное количество дней. Перегрузить операцию « \leftrightarrow » для нахождения разности дат и операции сравнения. Для класса определить оператор « $<<$ ». Стандартные функции и типы C для работы с датами не использовать.

2. Создать класс для хранения строк. Запрограммировать методы поиска подстроки, копирования, замены и удаления заданной подстроки, определения длины строки. Перегрузить операцию « $+$ » для конкатенации строк, операцию присваивания и операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать char\&) с проверкой допустимости индекса.

Раздел 5.

1. Создать класс для хранения одномерных целочисленных массивов. Обеспечить возможность задания количества элементов и базовой индексации. Запрограммировать методы поиска элементов и сортировки. Перегрузить операции для сложения и вычитания векторов. Перегрузить операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать int\&) с проверкой допустимости индекса.

2. Создать класс для хранения обыкновенных дробей. Запрограммировать метод сокращения дроби. Перегрузить арифметические операции. Для класса определить оператор « $<<$ ». Предусмотреть возбуждение исключительных ситуаций (при делении на ноль, переполнении).

Раздел 6.

Цель проекта: разработать программу «Текстовый редактор» со стандартным функционалом (копирование, открытие файла, работа со шрифтом).

1 Создайте графическое приложение **Qt Widgets**. Имя класса *notepad*.

2 Разработать интерфейс приложения, как показано на рисунке 8.1.

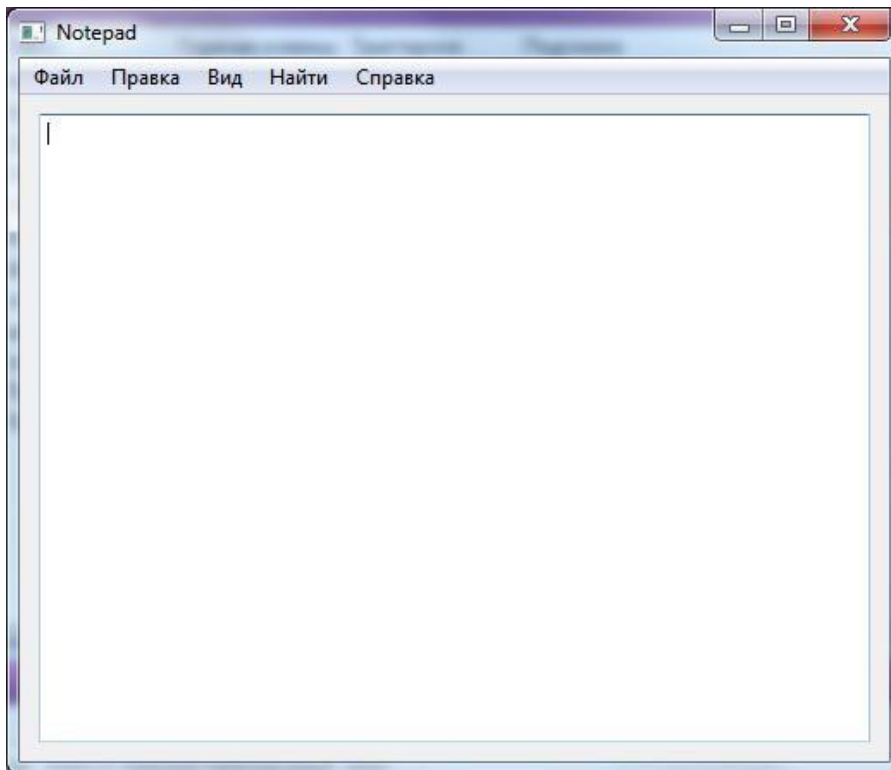


Рисунок 8.1 – Форма проекта *notepad*

3 Разработать меню. Пункты меню:

- 1) Файл (Новый, Открыть, Сохранить как);
- 2) Правка (Отменить, Копировать, Вставить);
- 3) Вид (Шрифт, Выравнивание (Слева, По центру, Справа));
- 4) Найти (Поиск, Перейти к строке...);
- 5) Справка.

Виджет `QTextEdit` сконфигурирован «по горизонтали». Поставьте запрет на изменение размера окна.

4 Переход к слотам событий: `action_ → Перейти к слоту → triggered()`.

Перейдем к слоту, который отвечает за открытие нового документа:

```
ui→textEdit→clear();
```

5 Перейдем к слоту, который отвечает за открытие ранее сохраненного документа:

```
QFileDialog dialog;  
dialog.setFileMode(QFileDialog::AnyFile);  
QString filename = dialog.getOpenFileName(NULL, "Open file", "", "");  
QFile file(filename);  
file.open(QFile::ReadOnly|QFile::Text);  
QTextStream in(&file);  
ui→textEdit→setText(in.readAll());  
file.close();
```

Класс `QFileDialog` – стандартное диалоговое окно, которое позволяет пользователям выбирать файлы или каталоги.

`FileMode` используется для указания того, что пользователь может выбрать в данном окне, в нашем случае *Имя файла*, если файл будет не найден, появится соответствующее информационное сообщение.

GetOpenFileName – удобная статистическая функция, которая возвращает существующий файл, выбранный пользователем. В параметрах указано, что заголовок окна «**Open File**», пустые кавычки обозначают, что пользователь может выбрать/найти файл в любом каталоге и с любым расширением.

В методах открытия файла перечислены режимы, в которых происходит открытие (для чтения и при окончании линии, переход на другую строку).

6 Перейдем к слоту, который отвечает за сохранение документа и выбора каталога для его сохранения:

```
QFileDialog dialog;  
dialog.setFileMode(QFileDialog::AnyFile);  
QString filename = dialog.getSaveFileName(NULL, "Save file", "", "");  
QFile file(filename);  
file.open(QFile::WriteOnly|QFile::Text);  
QTextStream out(&file);  
out <<  
ui->textEdit->toHtml();  
out.flush();  
file.close();
```

getSaveFileName - это статическая функция, которая возвращает имя файла, выбранного пользователем. Файл не должен существовать.

Функция класса **QTextEdit** - **toHTML** возвращает текст как **html**, т.е. сохраняет все элементы форматирования.

7 Перейдем к слоту, который отвечает за отмену действия:

```
ui->textEdit->undo();
```

8 Перейдем к слотам, которые отвечают за копирование и вставку выделенного текста:

```
_copytext = ui->textEdit->textCursor().selectedText();
```

Объявим в секции **private** переменную строкового типа **_copytext**. Заносим в неё выделенный текст.

```
ui->textEdit->textCursor().insertHtml(_copytext);
```

Вставляем текст, сохраняя все элементы **HTML** форматирования с текущей позиции курсора.

9 Перейдем к слотам, которые отвечают за выравнивание текста:

```
слева ui->textEdit->setAlignment(Qt::AlignLeft);
```

```
по центру ui->textEdit->setAlignment(Qt::AlignCenter);
```

```
справа ui->textEdit->setAlignment(Qt::AlignRight);
```

10 Перейдем к слоту, который отвечает за изменение шрифта в тексте:

```
QFontDialog dialog;  
QFont font =  
dialog.getFont(NULL);  
QTextCharFormat format;  
format.setFont(font);  
ui->textEdit->textCursor().setCharFormat(format);
```

Класс **QFont** определяет шрифт, используемый для форматирования текста. Функция **getFont()** открывает модальный диалог выбора шрифта и возвращает шрифт.

QTextCharFormat класс предоставляет информацию о форматировании для символов. На формат символов текста в документе указывают визуальные свойства текста.

Функция **setFont()** устанавливает выбранный шрифт.

11 Перейдем к слоту, который отвечает за переход на нужную строку:

```
bool ok;  
int num = QInputDialog::getInt(0, "Перейти", "На строку:", 1, 1,  
ui->textEdit->document()->lineCount(), 1, &ok);
```

```
QTextBlock block = ui->textEdit->document()->findBlockByLineNumber(num-1);
QTextCursor cursor(block);
ui->textEdit->setTextCursor(cursor);
```

[QInputDialog](#) класс, который предоставляет простой диалог для получения одного значения от пользователя.

Функция **getInt()** возвращает целое число, которое было введено пользователем. Параметры («Перейти» это текст, который отображается в строке заголовка диалогового окна. «На строку» – текст, который показывается пользователю (он должен сказать, что должно быть введено). 1 – является целым числом, по умолчанию с которого счётчик будет установлен. 1 – минимальное значение, которое пользователь может выбрать. Функция **lineCount()** считает количество строк в документе и является максимальным значением, которое пользователь может выбрать. 1 – сумма, на которую значения изменяются при нажатии пользователем кнопки со стрелками для увеличения или уменьшения значения.

[QTextBlock](#) класс предоставляет контейнер для фрагментов текста.

Функция **findBlockByLineNumber(LineNumber)** возвращает текстовый блок, который содержит указанный **LineNumber**.

12 Перейдем к слоту, который отвечает за поиск слова в тексте:

```
bool bOk;
QString str = QInputDialog::getText(0,"Найти","Введите текст:",
                                   QLineEdit::Normal,"",&bOk);

if (!bOk) { return; }
QTextCursor oldCursor =
ui->textEdit->textCursor();
ui->textEdit->setTextCursor(
    QTextCursor(ui->textEdit->document()->findBlockByLineNumber(0)));
bool finded = ui->textEdit->find(str);
if (!finded) ui->textEdit->setTextCursor(oldCursor);
```

Самостоятельно доработайте проект, добавив в него следующие пункты главного меню: Файл (Сохранить, Печать, Выход); Правка (Повторить, Вырезать, Удалить, Вставить дату); Найти (Заменить); Справка (О программе).

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|--|--|---|
| ОПК-2: Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности | | |
| ОПК-2.1 | Применяет современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности | <p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Элементы языка Си. Константы, идентификаторы, ключевые слова. 2. Типы данных и их объявление. Целые и вещественные типы. Перечисляемый тип. 3. Типы данных и их объявление. Указатели. Операции разадресации и адреса. Адресная арифметика. 4. Выражения. Операнды и операции (унарные, бинарные, тернарные). Правила преобразования типов. 5. Операторы языка Си. Оператор выражение, составной оператор, операторы условного перехода. 6. Организация циклических вычислительных процессов с помощью операторов for, while, do while. 7. Организация ввода-вывода в языке Си. Форматный ввод-вывод. 8. Массивы. Индексные выражения. Хранение в памяти одномерных и многомерных массивов. 9. Массивы. Основные алгоритмы их обработки. Ввод-вывод, поиск экстремума, сортировка. 10. Структуры и объединения. Вариантные структуры. Поля битов. 11. Правила определения переменных и типов. Инициализация данных. 12. Определение и вызов функций. Фактические и формальные параметры. 13. Определение и вызов функций. Передача массивов и указателей на функции. 14. Определение и вызов функций. |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>Предварительная инициализация параметров, функции с переменным числом параметров. Передача параметров функции main.</p> <p>15. Время жизни и область видимости программных объектов. Классы памяти. Инициализация глобальных и локальных переменных</p> <p>16. Динамические объекты. Способы выделения и освобождения памяти. Линейный односвязный список.</p> <p>17. Динамические массивы. Особенности выделения и освобождения памяти для многомерных массивов.</p> <p>18. Директивы препроцессора. Макроопределения.</p> <p>19. Объектно-ориентированный подход к программированию. Классы.</p> <p>20. Объектно-ориентированный подход к программированию. Инициализация и разрушение объектов. Конструкторы и деструкторы.</p> <p>21. Объектно-ориентированный подход к программированию. Ограничения доступа к членам класса. Друзья класса.</p> <p>22. Объектно-ориентированный подход к программированию. Наследование.</p> <p>23. Перегрузка операций.</p> <p>24. Организация ввода-вывода на языке C++. Потоки ввода-вывода.</p> <p>25. Шаблоны функций.</p> <p>26. Шаблоны классов.</p> <p>27. Библиотека STL. Другие библиотеки контейнерных классов.</p> <p>28. Обработка исключительных ситуаций <i>Практические задания</i></p> <p>1. Создать класс для хранения температур в шкалах Цельсия и Фаренгейта (при переводе из шкалы Цельсия в шкалу Фаренгейта исходное число умножают на 9/5 и прибавляют 32). Для класса определить арифметические операции и операцию <<.</p> <p>2. Создать класс для хранения масс тел в граммах, килограммах и тоннах. Для класса определить арифметические операции и</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|--|
| | | <p>операцию <<.</p> <ol style="list-style-type: none"> 3. Перегрузить операцию «*» для класса matrix, осуществляющую перемножение матриц. В случае невозможности этой операции — возбудить исключение. 4. Создать функцию, осуществляющую вычисление корней квадратного уравнения. В случае невозможности этой операции — возбудить исключение. 5. Создать класс для хранения целых чисел в двоичной и десятичной системах счисления. Для класса определить арифметические операции и операцию <<. 6. Создать класс для хранения множества простых чисел заданного диапазона. Определить методы проверки принадлежности заданного числа диапазону, определения количества чисел, получения предыдущего и следующего числа и т.д. Для класса определить операцию <<. 7. Создать шаблонную функцию для нахождения всех индексов минимальных (максимальных) элементов одномерного массива. 8. Создать класс для генерации целых и вещественных чисел из заданного диапазона. 9. Создать функцию, осуществляющую нахождение корней системы двух линейных уравнений (два неизвестных). В случае невозможности этой операции — возбудить исключение. 10. Создать функцию для вычисления значения определенного интеграла методом прямоугольников. Одним из параметров этой функции должен быть указатель на подинтегральную функцию. 11. Создать функцию для вычисления значения корня нелинейного уравнения $f(x)=0$ методом половинного деления. Одним из параметров этой функции должен быть указатель на функцию $f(x)$. 12. Создать функцию для вычисления значения экстремума нелинейного уравнения $f(x)=0$ методом деления на три части. Одним из параметров этой функции должен быть указатель на функцию $f(x)$. 13. Создать класс для хранения скоростей тел в м/с и км/ч. Для класса определить |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>арифметические операции и операцию <<.</p> <p>14. Создать класс для хранения множества треугольников. Треугольники задаются координатами вершин. Если заданный треугольник не существует — возбудить исключение. Разработать методы подсчета площади и периметра.</p> <p>15. Создать класс для хранения календарных дат. Обеспечить возможность работы с датами в различных форматах, изменения даты на заданное количество дней. Перегрузить операцию «-» для нахождения разности дат и операции сравнения. Для класса определить оператор <<. Стандартные функции и типы C для работы с датами не использовать.</p> <p>16. Создать класс для хранения строк. Запрограммировать методы поиска подстроки, копирования, замены и удаления заданной подстроки, определения длины строки. Перегрузить операцию «+» для конкатенации строк, операцию присваивания и операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать char&) с проверкой допустимости индекса.</p> <p>17. Создать класс для хранения одномерных целочисленных массивов. Обеспечить возможность задания количества элементов и базовой индексации. Запрограммировать методы поиска элементов и сортировки. Перегрузить операции для сложения и вычитания векторов. Перегрузить операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать int&) с проверкой допустимости индекса.</p> <p>18. Создать класс для хранения обыкновенных дробей. Запрограммировать метод</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|---|---|---|
| | | сокращения дроби. Перегрузить арифметические операции. Для класса определить оператор <<. Предусмотреть возбуждение исключительных ситуаций (при делении на ноль, переполнении) |
| ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения | | |
| ОПК-7.1 | Определяет средства разработки программных средств для решения практических задач профессиональной деятельности | <p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Элементы языка Си. Константы, идентификаторы, ключевые слова. 2. Типы данных и их объявление. Целые и вещественные типы. Перечисляемый тип. 3. Типы данных и их объявление. Указатели. Операции разадресации и адреса. Адресная арифметика. 4. Выражения. Операнды и операции (унарные, бинарные, тернарные). Правила преобразования типов. 5. Операторы языка Си. Оператор выражение, составной оператор, операторы условного перехода. 6. Организация циклических вычислительных процессов с помощью операторов for, while, do while. 7. Организация ввода-вывода в языке Си. Форматный ввод-вывод. 8. Массивы. Индексные выражения. Хранение в памяти одномерных и многомерных массивов. 9. Массивы. Основные алгоритмы их обработки. Ввод-вывод, поиск экстремума, сортировка. 10. Структуры и объединения. Вариантные структуры. Поля битов. 11. Правила определения переменных и типов. Инициализация данных. 12. Определение и вызов функций. Фактические и формальные параметры. 13. Определение и вызов функций. Передача массивов и указателей на функции. 14. Определение и вызов функций. |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>Предварительная инициализация параметров, функции с переменным числом параметров. Передача параметров функции main.</p> <p>15. Время жизни и область видимости программных объектов. Классы памяти. Инициализация глобальных и локальных переменных</p> <p>16. Динамические объекты. Способы выделения и освобождения памяти. Линейный односвязный список.</p> <p>17. Динамические массивы. Особенности выделения и освобождения памяти для многомерных массивов.</p> <p>18. Директивы препроцессора. Макроопределения.</p> <p>19. Объектно-ориентированный подход к программированию. Классы.</p> <p>20. Объектно-ориентированный подход к программированию. Инициализация и разрушение объектов. Конструкторы и деструкторы.</p> <p>21. Объектно-ориентированный подход к программированию. Ограничения доступа к членам класса. Друзья класса.</p> <p>22. Объектно-ориентированный подход к программированию. Наследование.</p> <p>23. Перегрузка операций.</p> <p>24. Организация ввода-вывода на языке C++. Потоки ввода-вывода.</p> <p>25. Шаблоны функций.</p> <p>26. Шаблоны классов.</p> <p>27. Библиотека STL. Другие библиотеки контейнерных классов.</p> <p>28. Обработка исключительных ситуаций</p> <p><i>Практические задания</i></p> <p>1. Создать класс для хранения температур в шкалах Цельсия и Фаренгейта (при переводе из шкалы Цельсия в шкалу Фаренгейта исходное число умножают на 9/5 и прибавляют 32). Для класса определить арифметические операции и операцию <<.</p> <p>2. Создать класс для хранения масс тел в</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|--|
| | | <p>граммах, килограммах и тоннах. Для класса определить арифметические операции и операцию \ll.</p> <ol style="list-style-type: none"> 3. Перегрузить операцию «*» для класса <code>matrix</code>, осуществляющую перемножение матриц. В случае невозможности этой операции — возбудить исключение. 4. Создать функцию, осуществляющую вычисление корней квадратного уравнения. В случае невозможности этой операции — возбудить исключение. 5. Создать класс для хранения целых чисел в двоичной и десятичной системах счисления. Для класса определить арифметические операции и операцию \ll. 6. Создать класс для хранения множества простых чисел заданного диапазона. Определить методы проверки принадлежности заданного числа диапазону, определения количества чисел, получения предыдущего и следующего числа и т.д. Для класса определить операцию \ll. 7. Создать шаблонную функцию для нахождения всех индексов минимальных (максимальных) элементов одномерного массива. 8. Создать класс для генерации целых и вещественных чисел из заданного диапазона. 9. Создать функцию, осуществляющую нахождение корней системы двух линейных уравнений (два неизвестных). В случае невозможности этой операции — возбудить исключение. 10. Создать функцию для вычисления значения определенного интеграла методом прямоугольников. Одним из параметров этой функции должен быть указатель на подинтегральную функцию. 11. Создать функцию для вычисления значения корня нелинейного уравнения $f(x)=0$ методом половинного деления. Одним из параметров этой функции должен быть указатель на функцию $f(x)$. 12. Создать функцию для вычисления значения экстремума нелинейного уравнения $f(x)=0$ методом деления на три части. Одним из параметров этой функции должен быть указатель на функцию $f(x)$. |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>13. Создать класс для хранения скоростей тел в м/с и км/ч. Для класса определить арифметические операции и операцию <<.</p> <p>14. Создать класс для хранения множества треугольников. Треугольники задаются координатами вершин. Если заданный треугольник не существует — возбудить исключение. Разработать методы подсчета площади и периметра.</p> <p>15. Создать класс для хранения календарных дат. Обеспечить возможность работы с датами в различных форматах, изменения даты на заданное количество дней. Перегрузить операцию «-» для нахождения разности дат и операции сравнения. Для класса определить оператор <<. Стандартные функции и типы C для работы с датами не использовать.</p> <p>16. Создать класс для хранения строк. Запрограммировать методы поиска подстроки, копирования, замены и удаления заданной подстроки, определения длины строки. Перегрузить операцию «+» для конкатенации строк, операцию присваивания и операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать char&) с проверкой допустимости индекса.</p> <p>17. Создать класс для хранения одномерных целочисленных массивов. Обеспечить возможность задания количества элементов и базовой индексации. Запрограммировать методы поиска элементов и сортировки. Перегрузить операции для сложения и вычитания векторов. Перегрузить операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать int&) с проверкой допустимости индекса.</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|--|---|
| | | 18. Создать класс для хранения обыкновенных дробей. Запрограммировать метод сокращения дроби. Перегрузить арифметические операции. Для класса определить оператор <<. Предусмотреть возбуждение исключительных ситуаций (при делении на ноль, переполнении) |
| ОПК-7.2 | Разрабатывает алгоритмы и программы для решения прикладных задач различных классов | <p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Элементы языка Си. Константы, идентификаторы, ключевые слова. 2. Типы данных и их объявление. Целые и вещественные типы. Перечисляемый тип. 3. Типы данных и их объявление. Указатели. Операции разадресации и адреса. Адресная арифметика. 4. Выражения. Операнды и операции (унарные, бинарные, тернарные). Правила преобразования типов. 5. Операторы языка Си. Оператор выражение, составной оператор, операторы условного перехода. 6. Организация циклических вычислительных процессов с помощью операторов for, while, do while. 7. Организация ввода-вывода в языке Си. Форматный ввод-вывод. 8. Массивы. Индексные выражения. Хранение в памяти одномерных и многомерных массивов. 9. Массивы. Основные алгоритмы их обработки. Ввод-вывод, поиск экстремума, сортировка. 10. Структуры и объединения. Вариантные структуры. Поля битов. 11. Правила определения переменных и типов. Инициализация данных. 12. Определение и вызов функций. Фактические и формальные параметры. 13. Определение и вызов функций. Передача массивов и указателей на функции. 14. Определение и вызов функций. Предварительная инициализация параметров, функции с переменным числом параметров. Передача параметров функции |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>main.</p> <ol style="list-style-type: none"> 15. Время жизни и область видимости программных объектов. Классы памяти. Инициализация глобальных и локальных переменных 16. Динамические объекты. Способы выделения и освобождения памяти. Линейный односвязный список. 17. Динамические массивы. Особенности выделения и освобождения памяти для многомерных массивов. 18. Директивы препроцессора. Макроопределения. 19. Объектно-ориентированный подход к программированию. Классы. 20. Объектно-ориентированный подход к программированию. Инициализация и разрушение объектов. Конструкторы и деструкторы. 21. Объектно-ориентированный подход к программированию. Ограничения доступа к членам класса. Друзья класса. 22. Объектно-ориентированный подход к программированию. Наследование. 23. Перегрузка операций. 24. Организация ввода-вывода на языке C++. Потоки ввода-вывода. 25. Шаблоны функций. 26. Шаблоны классов. 27. Библиотека STL. Другие библиотеки контейнерных классов. 28. Обработка исключительных ситуаций <p><i>Практические задания</i></p> <ol style="list-style-type: none"> 1. Создать класс для хранения температур в шкалах Цельсия и Фаренгейта (при переводе из шкалы Цельсия в шкалу Фаренгейта исходное число умножают на 9/5 и прибавляют 32). Для класса определить арифметические операции и операцию <<. 2. Создать класс для хранения масс тел в граммах, килограммах и тоннах. Для класса определить арифметические операции и операцию <<. 3. Перегрузить операцию «*» для класса matrix, осуществляющую перемножение |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>матриц. В случае невозможности этой операции — возбудить исключение.</p> <ol style="list-style-type: none"> 4. Создать функцию, осуществляющую вычисление корней квадратного уравнения. В случае невозможности этой операции — возбудить исключение. 5. Создать класс для хранения целых чисел в двоичной и десятичной системах счисления. Для класса определить арифметические операции и операцию \ll. 6. Создать класс для хранения множества простых чисел заданного диапазона. Определить методы проверки принадлежности заданного числа диапазону, определения количества чисел, получения предыдущего и следующего числа и т.д. Для класса определить операцию \ll. 7. Создать шаблонную функцию для нахождения всех индексов минимальных (максимальных) элементов одномерного массива. 8. Создать класс для генерации целых и вещественных чисел из заданного диапазона. 9. Создать функцию, осуществляющую нахождение корней системы двух линейных уравнений (два неизвестных). В случае невозможности этой операции — возбудить исключение. 10. Создать функцию для вычисления значения определенного интеграла методом прямоугольников. Одним из параметров этой функции должен быть указатель на подинтегральную функцию. 11. Создать функцию для вычисления значения корня нелинейного уравнения $f(x)=0$ методом половинного деления. Одним из параметров этой функции должен быть указатель на функцию $f(x)$. 12. Создать функцию для вычисления значения экстремума нелинейного уравнения $f(x)=0$ методом деления на три части. Одним из параметров этой функции должен быть указатель на функцию $f(x)$. 13. Создать класс для хранения скоростей тел в м/с и км/ч. Для класса определить арифметические операции и операцию \ll. 14. Создать класс для хранения множества треугольников. Треугольники задаются |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>координатами вершин. Если заданный треугольник не существует — возбудить исключение. Разработать методы подсчета площади и периметра.</p> <p>15. Создать класс для хранения календарных дат. Обеспечить возможность работы с датами в различных форматах, изменения даты на заданное количество дней. Перегрузить операцию «-» для нахождения разности дат и операции сравнения. Для класса определить оператор <<. Стандартные функции и типы C для работы с датами не использовать.</p> <p>16. Создать класс для хранения строк. Запрограммировать методы поиска подстроки, копирования, замены и удаления заданной подстроки, определения длины строки. Перегрузить операцию «+» для конкатенации строк, операцию присваивания и операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать char&) с проверкой допустимости индекса.</p> <p>17. Создать класс для хранения одномерных целочисленных массивов. Обеспечить возможность задания количества элементов и базовой индексации. Запрограммировать методы поиска элементов и сортировки. Перегрузить операции для сложения и вычитания векторов. Перегрузить операцию индексирования (т.к. оператор взятия индекса может появляться как слева, так и справа от оператора присваивания, то функция должна возвращать int&) с проверкой допустимости индекса.</p> <p>18. Создать класс для хранения обыкновенных дробей. Запрограммировать метод сокращения дроби. Перегрузить арифметические операции. Для класса</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>определить оператор <<. Предусмотреть возбуждение исключительных ситуаций (при делении на ноль, переполнении)</p> <p>Разработка индивидуального проекта по одной из тем:</p> <p>Игры</p> <ol style="list-style-type: none"> 1. "Сапер" 2. "Змейка" - 3. "Тетрис" - 4. "Три в ряд" 5. "Тамагочи" 6. "Пятнашки" 7. "Морской бой" . 8. "Мемогу" ("парные картинки") 9. Пазл <p>Полезные приложения</p> <ol style="list-style-type: none"> 1. Текстовый редактор ("Блокнот") - 2. Графический редактор ("Paint") - 3. "Стикерс" 4. Работа с кодировками ("Штирлиц") <p>Учётные задачи</p> <ol style="list-style-type: none"> 1. Учёт инвентаря 2. Система бронирования 3. Расписание приема врача <p>Работа с данными</p> <ol style="list-style-type: none"> 1. Менеджер кулинарных рецептов 2. Галерея изображений 3. Словарь логопеда 4. Семейное древо 5. Викторина <p>Веб</p> <ol style="list-style-type: none"> 1. Чат 2. Доска для рисования онлайн 3. Генератор поздравительных открыток 4. CAPTCHA-генератор |