Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 18.08.2025 18:24:39
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____/ Д.Г. Демидов /
«_27_» ____февраля_____ 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Программирование и основы алгоритмизации систем управления

Направление подготовки
09.03.01 «Информатика и вычислительная техника»

Профиль подготовки (образовательная программа)
«Интеллектуальные беспилотные системы»

Год начала обучения:
2025

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Очная

Москва – 2025

Рабочая программа составлена на основе ФГОС ВО и учебного плана Московского политехнического университета по направлению (специальности) 09.03.01 Информатики и вычислительная техника, по профилю подготовки Интеллектуальные беспилотные системы

Составитель рабочей программы:

доцент кафедры «СМАРТ технологии», к.т.н., доцент		Д.И.Давлетчин
(должность, ученое звание, степень)	(подпись)	(Ф.И.О.)

Рабочая программа утверждена на заседании кафедры

СМАРТ технологии
(наименование кафедры)

Заведующий кафедрой к.т.н., доцент		Е.В. Петрунина
	(подпись)	(Ф.И.О.)

СОГЛАСОВАНО:

Заведующий кафедрой «СМАРТ технологии», к.т.н., доцент		Е.В. Петрунина
	(подпись)	(Ф.И.О.)

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
2. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования
3. Структура и содержание дисциплины
3.1 Виды учебной работы и трудоемкость (по формам обучения)
3.1.1 Очная форма обучения
3.2 Тематический план изучения дисциплины (по формам обучения)
3.2.1 Очная форма обучения
3.3 Содержание дисциплины
3.4 Тематика семинарских/практических и лабораторных занятий
3.5 Тематика вопросов для самостоятельного изучения
4. Учебно-методическое и информационное обеспечение
4.1 Основная литература
4.2 Дополнительная литература
4.3 Электронные образовательные ресурсы
4.4 Лицензионное и свободно распространяемое программное обеспечение
4.5 Современные профессиональные базы данных и информационные справочные системы
5 Материально-техническое обеспечение
6 Методические рекомендации
6.1 Методические рекомендации для преподавателя по организации обучения
6.2 Методические указания для обучающихся по освоению дисциплины
6.3 Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
7 Фонд оценочных средств
7.1 Методы контроля и оценивания результатов обучения
7.2 Шкала и критерии оценивания результатов обучения
7.3 Оценочные средства
7.3.1 Текущий контроль на лабораторных занятиях
7.3.2 Промежуточная аттестация (экзамен)

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Выпускник, освоивший дисциплину, должен обладать следующими компетенциями:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Знает основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ОПК-8.2 Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ОПК-8.3 Владеет языком программирования, методами отладки и тестирования работоспособности программы

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование и основы алгоритмизации» относится к дисциплинам базовой части (Блока 1) основной образовательной программы бакалавриата; изучается в 1 семестре. Дисциплина базируется на следующих знаниях и навыках, приобретенных в области:

- «Программное обеспечение рабочего места оператора»;
- «Линейная алгебра».

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры
			1
1	Аудиторные занятия	64	64
	В том числе:		
1.1	Лекции	15	15
1.2	Семинарские/практические занятия		
1.3	Лабораторные занятия	48	48
2	Самостоятельная работа	80	80
	В том числе:		
2.1	Подготовка к практическим занятиям	70	70
2.2	Тестирование	10	10
3	Промежуточная аттестация		
	Экзамен		
	Итого:	144/4	144/4

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1	Основы языка С++. Переменные, константы, типы переменных		2		4		10
2	Программирование ветвящихся алгоритмов. Операторы условного перехода		2		4		10
3	Программирование циклических алгоритмов		2		4		10
4	Работа с указателями и массивами		2		4		10
5	Использование функций		2		8		10
6	Работа с файлами		2		8		10
7	Создание и использование динамически подключаемых библиотек		2		8		10
8	Основы объектно-ориентированного программирования. Классы.		2		8		10
Итого			16		48		80

3.3 Содержание дисциплины

Раздел 1. Основы языка С++. Переменные, константы, типы переменных.

• Язык Си++. Переменные, константы. Типы переменных. Опреторы объявления переменных.

• изучение пользовательского интерфейса MS Visual Studio и возможностей его конфигурации;

• изучение возможностей «мастера приложений» MS Visual Studio;

• изучение возможностей отладчика программ;

• стандартное пространство имен и библиотеки ввода-вывода;

• потоковый ввод/вывод.

Раздел 2. Программирование ветвящихся алгоритмов. Операторы условного перехода

• операторы условного перехода if/else;

• изучение синтаксиса операторов switch/case.

Раздел 3. Программирование циклических алгоритмов

• изучение синтаксиса операторов цикла for,

• изучение синтаксиса операторов while, do … while

• изучение применения break и continue в теле цикла

Раздел 4. Работа с указателями и массивами

• массивы

• изучение синтаксиса работы с указателями и массивами

• динамические массивы

• изучение синтаксиса операторов new, delete

• разыменование указателей

• изучение применения генератора псевдослучайных чисел
Раздел 5. Использование функций
• изучение синтаксиса объявления и реализации функций
• передача параметров по ссылке и по значению
• возвращаемое значение функции и оператор return

Раздел 6. Работа с файлами
• изучение синтаксиса функций работы с файлами
• текстовый режим и бинарный режимы работы с файлами
• режимы перезаписи и добавления данных
• функции работы с файлами

Раздел 7. Создание и использование динамически подключаемых библиотек
• изучение теории создания и использования DLL
• статическое и динамическое связывание
• экспорт и импорт функций из DLL

Раздел 8. Основы объектно-ориентированного программирования. Классы.
• основы объектно-ориентированного программирования. классы.
• конструкторы и деструктор класса
• модификаторы доступа к полям и методам класса
• синтаксис точка и стрелка
• использование заголовочных файлов

3.4 Тематика семинарских/практических и лабораторных занятий

Лабораторные занятия
Лабораторная работа №1. Знакомство с языком C++ IDE MS Visual Studio
Лабораторная работа №2. Программирование ветвящихся алгоритмов. Операторы условного перехода
Лабораторная работа №3. Программирование циклических алгоритмов
Лабораторная работа №4. Работа с динамическими массивами
Лабораторная работа №5. Использование функций
Лабораторная работа №6 Работа с файлами
Лабораторная работа №7. Создание и использование динамически подключаемых библиотек
Лабораторная работа №8. Основы ООП. Работа с классами

4. Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Лебеденко, Л. Ф. Основы программирования на C++ : учебное пособие : [16+] / Л. Ф. Лебеденко, О. И. Моренкова. – 2-е изд., перераб. и доп. – Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2021. – 200 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=694769 (дата обращения: 26.06.2023). – Библиогр. в кн. – Текст : электронный.
2. Литвиненко, В. А. Основы объектно-ориентированного программирования задач на графах : учебное пособие : [16+] / В. А. Литвиненко ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 7 133 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=612227 (дата обращения: 26.06.2023). –

Библиогр.: с. 99. – ISBN 978-5-9275-3472-2. – Текст : электронный.

3. Затонский, А. В. Программирование и основы алгоритмизации. Теоретические основы и примеры реализации численных методов: учебное пособие / А.В. Затонский, Н.В. Бильфельд. — 2-е изд. — Москва : РИОР : ИНФРА-М, 2020. — 167 с. — (Высшее образование). — DOI: https: //www.dx.doi.org/10.12737/20468. - ISBN 978-5-369-01195- 9. - Текст : электронный. - URL: https://znanium.com/catalog/product/1077389 (дата обращения: 26.06.2023). – Режим доступа: по подписке.
4. Павловская, Т. А. C/C++. Структурное и объектно-ориентированное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. - Санкт-Петербург : Питер, 2021. - 352 с. - (Серия «Учебное пособие»). - ISBN 978-5-4461-9799-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1857042 (дата обращения: 26.06.2023). – Режим доступа: по подписке.
5. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5- 8199-0699-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1172261 (дата обращения: 26.06.2023). – Режим доступа: по подписке.

4.2 Дополнительная литература

1. Дэвид Дж. Круглински, Скотт Уингоу, Дж. Шеферд. Программирование на Microsoft Visual C++ 6.0 для профессионалов. М.: Русская редакция, 2004 г.-861 с.
2. Александров, Э.Э. Программирование на языке C в Microsoft Visual Studio 2010 : учебное пособие / Э.Э. Александров, В.В. Афонин ; Национальный Открытый Университет "ИНТУИТ". – Москва : Интернет-Университет Информационных Технологий, 2010. – 500 с. : ил. –URL: http://biblioclub.ru/index.php?page=book&id=233564 (дата обращения: 26.06.2023). – Текст : электронный.

4.3 Электронные образовательные ресурсы

1. «Основы программирования» https://online.mospolytech.ru/course/view.php?id=2592

4.4 Лицензионное и свободно распространяемое программное обеспечение

1. Microsoft Visual Studio Community Edition с установленным компилятором C++

4.5 Современные профессиональные базы данных и информационные справочные системы

1. Справочно-правовая системы «КонсультантПлюс: Некоммерческая интернетверсия» https://www.consultant.ru/online/
2. Официальный интернет-портал правовой информации http://pravo.gov.ru
3. Российская национальная библиотека http://www.nlr.ru
4. ЭБС «Университетская библиотека онлайн» https://biblioclub.ru/index.php
5. Единое окно доступа к образовательным ресурсам Федеральный портал http://window.edu.ru
6. Научная электронная библиотека http://www.elibrary.ru
7. Российская государственная библиотека http://www.rsl.ru

5. Материально-техническое обеспечение

1. Компьютерные классы с оснащением: столы, стулья, аудиторная доска, использование переносного мультимедийного комплекса (переносной проектор, персональный ноутбук).
2. Персональные компьютеры, мониторы, мышки, клавиатуры. Рабочее место преподавателя: стол, стул.
3. Аудитория для самостоятельной работы.
4. Библиотека, читальный зал.

6. Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.
2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.
3. При организации и проведения экзаменов в практико-ориентированной форме следует использовать утвержденные кафедрой Методические рекомендации.

6.2 Методические указания для обучающихся по освоению дисциплины

Получение углубленных знаний по дисциплине достигается за счет активной самостоятельной работы обучающихся. Выделяемые часы целесообразно использовать для знакомства с учебной и научной литературой по проблемам дисциплины, анализа научных концепций.

Промежуточная аттестация обучающихся в форме зачѐта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «не удовлетворительно».

К промежуточной аттестации допускаются только обучающиеся, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Программирование и основы алгоритмизации систем управления».

6.3 Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с

ограниченными возможностями здоровья: - создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников, например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и ассимиляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи: - продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут; - продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления, обучающегося при защите курсовой работы - не более чем на 15 минут.

Для обеспечения подготовки людей в формате очной аудиторной работы с ограниченными возможностями движения выбираются аудитории с доступностью в рамках требований по организации безбарьерной среды движения.

7. Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ОПК-8 Способность разрабатывать алгоритмы и программы, пригодные для практического применения				
Показатель	Критерии оценивания			
	1	2	3	4
ОПК 1.1: - основы алгоритмизаци и (свойства алгоритмов, область применения алгоритмов); - методы построения алгоритмов; -	Обучающийс я демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: основ алгоритмизаци и;	Обучающийся демонстрирует неполное соответствие следующих знаний:основ алгоритмизации; методов построения	Обучающийся демонстрирует частичное соответствие следующих знаний: основ алгоритмизации; методов построения	Обучающийся демонстрирует полное соответствие следующих знаний: основ алгоритмизации; методов построения

структуры данных; - синтаксис и семантику универсального алгоритмическ ого языка программирова ния высокого уровня; - основные принципы и методологию разработки прикладного программного обеспечения; - типовые способы организации программных данных; - подходы к построению программных алгоритмов.	методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмическ ого языка программирова ния высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов.	алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмическог о языка программирован ия высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации	алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмическог о языка программирован ия высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмическог о языка программирован ия высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Свободно оперирует приобретенными знаниями.
ОПК 1.2: - строить блоксхемы алгоритмов; - проводить анализ эффективности алгоритмов; - описать алгоритм, используя ключевые слова языков программирова ния, но опуская подробности и специфический синтаксис (псевдокод).	Обучающийс я не умеет или в недостаточной степени умеет: осуществлять синтез - строить блоксхемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирова ния, но опуская подробности и специфический синтаксис (псевдокод).	Обучающийся демонстрирует неполное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирован ия, но опуская подробности и специфический синтаксис (псевдокод).	Обучающийся демонстрирует частичное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирован ия, но опуская подробности и специфический синтаксис (псевдокод).	Обучающийся демонстрирует полное соответствие следующих умений:осуществ лять синтез - строить блоксхемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирован ия, но опуская подробности и специфический синтаксис (псевдокод).

ОПК 1.3: - навыками составления алгоритмов и их представления в виде блоксхем; - навыками формализации прикладных задач.	Обучающийс я не владеет или в недостаточной степени владеет навыками составления алгоритмов и их представления в виде блоксхем; навыками формализации прикладных задач	Обучающийся владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач.	Обучающийся частично владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач	Обучающийся в полном объеме владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач.

7.2 Шкала и критерии оценивания результатов обучения

Критерии оценки ответа на экзамене

«отлично»: обучающийся демонстрирует системные теоретические знания, практические навыки, владеет терминами, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью и способность быстро реагировать на уточняющие вопросы.

«хорошо»: обучающийся демонстрирует прочные теоретические знания, практические навыки, владеет терминами, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью, но при этом делает несущественные ошибки, которые быстро исправляет самостоятельно или при незначительной коррекции преподавателем.

«удовлетворительно»: обучающийся демонстрирует неглубокие теоретические знания, проявляет слабо сформированные навыки анализа явлений и процессов, недостаточное умение делать аргументированные выводы и приводить примеры, показывает не достаточно свободное владение монологической речью, терминами, логичностью и последовательностью изложения, делает ошибки, которые может исправить только при коррекции преподавателем.

«не удовлетворительно»: обучающийся демонстрирует незнание теоретических основ предмета, отсутствие практических навыков, не умеет делать аргументированные выводы и приводить примеры, показывает слабое владение монологической речью, не владеет терминами, проявляет отсутствие логичности и последовательности изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на дополнительные вопросы.

Критерии оценки работы обучающегося на лабораторных занятиях

«5» (отлично): выполнены все практические задания, предусмотренные практическими занятиями, обучающийся четко и без ошибок ответил на все контрольные вопросы, активно работал на практических занятиях.

«4» (хорошо): выполнены все практические задания, предусмотренные практическими занятиями, обучающийся с корректирующими замечаниями преподавателя ответил на все контрольные вопросы, достаточно активно работал на практических занятиях.

«3» (удовлетворительно): выполнены все практические задания, предусмотренные практическими занятиями с замечаниями преподавателя; обучающийся ответил на все контрольные вопросы с замечаниями.

«2» (неудовлетворительно): обучающийся не выполнил или выполнил неправильно практические задания, предусмотренные практическими занятиями; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

Критерии оценки тестирования

Тестирование оценивается в соответствии с процентом правильных ответов, данных студентом на вопросы теста. Стандартная шкала соответствия результатов тестирования выставляемой балльной оценке:

− «отлично» - свыше 85% правильных ответов;

− «хорошо» - от 70,1% до 85% правильных ответов;

− «удовлетворительно» - от 55,1% до 70% правильных ответов;

− от 0 до 55% правильных ответов – «неудовлетворительно»

7.3 Оценочные средства

7.3.1 Текущий контроль на лабораторных занятиях

Пример задания текущего контроля:

В каком порядке, эти два параметра, объявлены в функции main? Параметры: argc и argv

A) массив аргументов, затем количество элементов

B) Существует только один аргумент

C) они не объявлены в main

D) количество аргументов, затем массив аргументов

В каком случае можно не использовать фигурные скобочки в операторе выбора if?

A) если в теле оператора if всего один оператор

B) если в теле оператора if нет ни одного оператора

C) нет правильного ответа

D) если в теле оператора if два и более операторов

В чем различие использования следующих выражений #include <…> и #include «…»:

A) различие заключается в методе поиска препроцессором включаемого файла

B) в различии использования заголовочных и исходных файлов

C) нет различий

В чем различия между delete и delete[]?

A) Нет разницы

B) delete предназначен для уничтожения объектов, память под которые выделена при помощи new(). delete[] для объектов выделенных при помощи оператора new[]().

C) delete предназначен для уничтожения объектов, память под которые выделена при помощи new[](). delete[] для объектов выделенных при помощи оператора new().

В языке C++ встроенный тип данных «char» предназначен для хранения

A) целых чисел или символов;

B) вещественных чисел;

C) символов;

В языке C++ встроенный тип данных «double» предназначен для хранения

A) символов;

B) вещественных чисел;

C) целых чисел.

В языке C++ строковые литеральные константы представляют собой

A) одиночный символ, заключенный в апострофы, например 'w', 'g' или '7';

B) последовательность символов, заключенную в двойные кавычки, например "Это строка";

C) последовательность цифр, не начинающуюся с 0, например 23, 2003.

В языке C++ существует специальный оператор прерывания циклов break. Он служит для того, чтобы

A) досрочно прекратить выполнение содержащего его ближайшего цикла while, do … while или for или условного оператора switch.

B) досрочно прекратить выполнение текущей итерации содержащего его ближайшего цикла while, do … while или for.

C) досрочно завершить программу.

Выберите правильное объявление производного класса

A) class ElectricDevices:: Devices;

B) class ElectricDevices: public class Devices;

C) class ElectricDevices: public Devices;

D) class ElectricDevices: class(Devices);

Выберите правильное утверждение:

A) целой константе можно присвоить целую переменную

B) целой переменной можно присвоить целую константу

C) целой константе можно присвоить вещественную переменную

Для запуска программы без отладки нужно нажать

A) Ctrl+Shift+B

B) Ctrl+B

C) F5

D) Ctrl+F5

E) Shift+F5

Для остановки отладки нужно нажать

A) Ctrl+Shift+B

B) Ctrl+B

C) F5

D) Ctrl+F5

E) Shift+F5

Для сборки программы без запуска отладки нужно нажать

A) Ctrl+Shift+B

B) Ctrl+B

C) F5

D) Ctrl+F5
E) Shift+F5

Для сборки программы и запуска отладки нужно нажать
A) Ctrl+Shift+B
B) Ctrl+B
C) F5
D) Ctrl+F5
E) Shift+F5

Если условие оператора if ложное, то:
A) выполняется тело оператора выбора
B) программа завершает работу
C) выполняется следующий оператор, сразу после оператора if

7.3.2 Промежуточная аттестация (экзамен)

Как обозначается переход на новую строку
A) \t
B) \n
C) \w

Как ограничить диапазон генерации псевдослучайных чисел
A) использовать %
B) использовать %%
C) использовать *
D) использовать &

Как организовать запись в файл?
A) a_file.printf("запись");
B) a_file
C) "запись";
D) a_file << "запись";
E) a_file.out("запись");

Какая "маска" ввода соответствует вещественному числу обычной точности
A) %i
B) %f
C) %lf
D) %s

Какая операция используется для доступа к полям структуры?
A) операция :
B) операция .
C) операция ,

D) операция ::

E) операция *

Какая переменная доступна только в методе, где она объявлена?

A) переменная - член

B) локальная переменная

C) переменная класса

D) внешняя переменная

Какая самая крупная структура среди перечисленных с точки зрения иерархии разработки на Visual Studio

A) решение

B) проект

C) консорциум

D) контейнер

Какая функция выполняет начальную инициализацию данных в классе?

A) конструктор

B) нет правильного ответа

C) деструктор

Какие из следующих утверждений верные?

A) константный метод объявляется с ключевым словом const после списка параметров.

B) константный метод может изменять значения полей класса.

C) константный метод может вызывать любые методы класса.

D) константный метод может вызываться только для константных объектов.

E) константный метод может изменять значения константных полей класса.

Какое выражение не содержит синтаксических ошибок:

A) (-0.18)*a)/(r-0.2*t)

B) (-0.18)*a)\(r-0.2*t))

C) -0.18*a/r-0.2*t

Какое значение должен возвращать деструктор?

A) деструкторы не возвращают значение

B) указатель на класс

C) код состояния о правильном удалении класса

D) объект класса.

Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?

A) 1

B) -1

C) Программа не возвращает значение.

D) 0

Какое из следующих утверждений относительно классов и структур истинно?

A) структура может включать в себя только простейшие типы данных (int, char и т.д.).

B) структура не может иметь конструкторов и деструкторов.

C) структура не может быть наследником класса.

D) поля класса закрыты по умолчанию.

E) класс не может быть наследником структуры.

Какое свойство ООП позволяет единообразно обращаться с разными объектами одной иерархии?

A) инкапсуляция

B) абстрагирование

C) наследование

D) полиморфизм

Какой из следующих классов обрабатывает процесс записи в файл?

A) другое

B) input_file

C) ofstream

D) ifstream

Какой из следующих логических операторов - логический оператор И?

A) |

B) |&

C) &

D) &&

Какой модификатор обозначает конец строки

A) endl

B) lineend

C) cr

D) ln

E) break

Какой режим конфигурации сборки проекта предназначен для передачи пользователю?

A) Release

B) Debug

C) Remix

D) Defrag

Какую библиотеку можно использовать для руссификации вывода в консоль

A) locale

B) ruschar

C) rusconsole

D) oem866

Какую команду можно использовать для останова программы

A) system("pause")

B) system("end")

C) system("clear")

D) system("stop")

E) system("cls")