

Учреждение образования «Полоцкий государственный университет»

УТВЕРЖДАЮ
Проректор по учебной работе
учреждения образования
«Полоцкий государственный
университет»



Н.А. Борейко

« 01 » 07 2021 г.

Регистрационный № УД- 195/21/уч.

Модуль «Программирование»

ПРОГРАММИРОВАНИЕ

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности
1-31 04 08 «Компьютерная физика»

2021 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 04 08-2018 и учебного плана по специальности 1-31 04 08 «Компьютерная физика». Регистрационный № 06-20/уч. ФКНиЭ от 28.12.2020 г.



СОСТАВИТЕЛИ:

Вабищевич Сергей Ананьевич, к.ф.-м.н., доцент, заведующий кафедрой физики учреждения образования «Полоцкий государственный университет»

Солдатенко Павел Николаевич, старший преподаватель кафедры физики учреждения образования «Полоцкий государственный университет»

РЕЦЕНЗЕНТЫ:

Богущ Рихард Петрович, к.т.н., доцент, заведующий кафедрой вычислительных систем и сетей

Глухова Татьяна Михайловна, генеральный директор ООО «КлаудТехнолоджи»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой физики учреждения образования «Полоцкий государственный университет»

(протокол № 12 от «04» 06 2021 г.);

Методической комиссией факультета компьютерных наук и электроники учреждения образования «Полоцкий государственный университет»

(протокол № 4 от «16» 06 2021 г.)

Научно-методическим советом учреждения образования «Полоцкий государственный университет»

(протокол № 5 от «01» 04 2021 г.)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа по учебной дисциплине «Программирование» направлена на подготовку специалистов, применяющих в своей профессиональной деятельности современные методы прикладного программирования.

Целями преподавания учебной дисциплины «Программирование» являются:

- освоение студентами знаний основных конструкций языков программирования C, C++;
- изучение основных структур данных и алгоритмов их обработки, используемых в практике современного программирования;
- овладение начальными навыками разработки программного обеспечения, работающего в консольном режиме под управлением операционных систем Windows и Linux.

Достижение поставленных целей предполагает решение следующих **задач:**

- формирование у студентов базовых понятий структурного программирования, способствующих развитию логики;
- рассмотрение, в рамках единого подхода, основных закономерностей протекания информационных процессов;
- обучение студентов навыкам разработки программного обеспечения;
- обучение студентов основам алгоритмизации задачи и поиска пути ее решения;
- обучение студентов элементарным алгоритмам обработки структурной информации.

В результате изучения учебной дисциплины «Программирование» формируется следующая базовая профессиональная компетенция:

БПК-3. Владеть основными понятиями базового курса информатики, теории алгоритмов, основными конструкциями алгоритмических языков, технологиями объектно-ориентированного программирования для решения задач прикладной физики, уметь разрабатывать программное обеспечение в средах быстрой разработки приложений.

В результате изучения учебной дисциплины студент должен:

знать:

- основные конструкции, функции стандартной библиотеки, элементы STL и технологии языка C++;

уметь:

- разрабатывать алгоритмы на языке C++ для решения задач представления и обработки структурированной информации;
- разрабатывать на языке C++ простые программы для платформ Windows и Linux

владеть:

- навыками разработки программного обеспечения, работающего под управлением операционных систем Windows и Linux.

Учебная дисциплина «Программирование» входит в модуль «Программирование» в подготовке специалистов по специальности 1-31 04 08

«Компьютерная физика». Изучается в рамках государственного компонента учреждения высшего образования.

Для изучения курса программирования необходимо знание следующих разделов математики:

- элементы линейной алгебры и аналитической геометрии;
- дифференциальное исчисление функций одной и нескольких переменных;
- исследование функций с помощью производных;
- определенный и неопределенный интегралы, криволинейные и кратные интегралы;
- элементы теории дифференциальных уравнений;
- векторный анализ и основные понятия теории поля;
- теория вероятностей и математическая статистика.

Знания, полученные при изучении курса «Программирование» послужат базой для дальнейшего изучения таких дисциплин, как «Программно-аппаратные интерфейсы информационных систем», «Объектно-ориентированное программирование».

Форма получения высшего образования – дневная.

В соответствии с учебным планом на изучение учебной дисциплины отводится:

Общее количество часов – 216, из них аудиторных -104 часа, из них лекции – 44 часа, лабораторные занятия - 60 часов.

Распределение аудиторного времени по видам занятий, курсам и семестрам:

1 курс 1 семестр - всего 108 часов (3 з.е.), из них аудиторных 54 часа, в том числе лекции – 18 часов, лабораторные занятия - 36 часов; самостоятельная работа студентов 54 часа.

1 курс 2 семестр - всего 108 часов (3 з.е.), из них аудиторных 50 часов, в том числе лекции – 26 часов, лабораторные занятия - 24 часа; самостоятельная работа студентов 58 часов.

Формы текущей аттестации по учебной дисциплине «Программирование»:

1 семестр – зачёт; 2 семестр – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема1. Аппаратные и программные средства электронно-вычислительной машины (ЭВМ)

Структура и принципы работы ЭВМ историческое развитие ЭВМ. Архитектура фон Неймана, процессор, основной цикл процессора, структура процессора, регистры, арифметико-логическое устройство, устройство управления, кэш-память. Структура машинной команды, система команд процессора. Концепция шины, шины адреса, данных, управления. Оперативная память, структура оперативной памяти, способы реализации оперативной памяти (динамическая, статическая). Понятие об адресном пространстве оперативной памяти. Пространство адресов ввода-вывода. Система прерываний, периферийные устройства, высокоскоростные устройства и режим прямого доступа к памяти (каналы Direct Memory Access). Контроллеры и драйверы. Низкоскоростные периферийные устройства, принципы работы клавиатуры и мыши. Постоянное запоминающее устройство, CMOS, программа начальной загрузки. Операционные системы. Операционные системы персональных ЭВМ. Структура операционной системы, ядро, BIOS, драйверы, командный процессор. Служебные программы и утилиты.

Тема2. Алгоритмические языки

Языки программирования. Машинный язык и язык Ассемблера. Языки высокого уровня. Классификация языков программирования. Парадигмы программирования: процедурное, структурное, абстракция данных, модульное программирование, объектно-ориентированное программирование, функциональное программирование. Сравнительный анализ современных языков программирования.

Тема3. Основные конструкции языков С, С++

Идентификаторы, константы, переменные, типизация, скалярные типы данных. Выражения, операнды и операции. Структура программы, блоки, вложенность блоков, модули, пространства имен, области видимости, элементарные директивы препроцессора. Структуры управления, условные операторы, циклические конструкции. Массивы, указатели, адресная арифметика, ссылки, преобразования типов, структуры и объединения. Функции, передача параметров в функции, параметры по умолчанию, функции с переменным числом параметров, рекурсия.

Тема4. Основные функции библиотеки, элементарный ввод-вывод

Функции преобразования типов, элементарные функции С для работы со строками, работа с динамически распределяемой памятью. С и С++ функции ввода-вывода, форматирование, функции работы с файлами и файловой системой, текстовый и бинарный режимы открытия файлов, дескрипторно- и потоково-ориентированный ввод-вывод.

Тема5. Алгоритмы и структуры данных

Понятие алгоритма. Базовые алгоритмы для обработки данных простых типов и структур данных: работа с массивами, сортировка и поиск. Вычислительная сложность алгоритмов. Рекурсивные алгоритмы, примеры рекурсивных алгоритмов. Динамически создаваемые объекты. Списки, очереди, стеки, деревья, строками.

Тема6. Разработка консольных приложений в ОС Windows и Linux

Этапы жизненного цикла программ. Разработка и отладка консольных C++ приложений в системе Microsoft Visual Studio. Разработка и отладка консольных приложений под ОС Linux с использованием среды разработки KDevelop.

Тема7. Основы объектно-ориентированного программирования

Классы и экземпляры, поля и методы, наследование. Полиморфизм и полиморфные функции. Модификаторы доступа к полям и методам. Статические поля и методы. Конструкторы и деструкторы. Конструкторы по умолчанию и копирования. Абстрактные классы. Виртуальные функции. Перегрузка операций, дружественные функции и классы. Множественное наследование, виртуальные базовые классы.

Тема8. Стандартная библиотека C++ и STL

Шаблоны типов, функций классов. Контейнер string. Работа со строками, инициализация, присваивание, копирование, проблемы. Контейнеры vector, list. Использование vector контейнера для работы с динамическими массивами. Ассоциативные контейнеры map, multimap, hash_map. Итераторы. Специальные контейнеры. Функторы. Интернационализация.

Тема9. C++: ввод-вывод, работа с файлами

Потоки. Ввод-вывод встроенных типов, состояния потока, форматирование, манипуляторы. Ввод-вывод пользовательских типов данных, перегрузка операций ввода-вывода.

Тема10. Обработка исключительных ситуаций в C++

Традиционный подход к обработке ошибок. Концепция исключения, перехват исключения, вложенная обработка исключений. Возбуждение исключения. Стандартные типы исключений, исключительные ситуации при работе с числами.

Тема11. Разработка приложений в ОС Windows и Linux

Этапы жизненного цикла программ. Разработка оконных C++ приложений в системе Microsoft Visual Studio. Разработка оконных приложений под ОС Linux.

1.2.	<p>Алгоритмические языки Лекции Языки программирования. Машинный язык и язык Ассемблера. Языки высокого уровня. Классификация языков программирования. Парадигмы программирования: процедурное, структурное, абстракция данных, модульное программирование, ООП, функциональное программирование. Сравнительный анализ современных языков программирования.</p> <p>Лабораторные занятия Оператор присваивания, арифметико-логические операции со встроенными типами данных. Элементарные функции ввода-вывода.</p>	2	4		Лекционная презентация № 1-2	[1], [4]	Защита отчета по лаб. работе
1.3.	<p>Основные конструкции языков C, C++. Лекции Идентификаторы, константы, переменные, типизация, скалярные типы данных. Выражения, операнды и операции. Структура программы, блоки, вложенность блоков, модули, пространства имен, области видимости, элементарные директивы препроцессора.</p> <p>Структуры управления, условные операторы, циклические конструкции. Массивы, указатели, адресная арифметика, ссылки, преобразования типов, структуры и объединения. Функции, передача параметров в функции, параметры по умолчанию, функции с переменным числом параметров, рекурсия..</p> <p>Лабораторное занятие Работа с основными конструкциями управления</p>	2 2	4		Лекционная презентация № 1-3	[1], [4]	Краткое сообщение по результатам углубленного изучения отдельных тем дисциплины. Защита отчета по лаб. работе
1.4.	<p>Основные функции библиотеки, элементарный ввод-вывод. Лекции Функции преобразования типов, элементарные функции C для работы со строками, работа с динамически распределяемой памятью. C и C++ функции ввода-вывода, форматирование, функции работы с файлами и файловой системой, текстовый и бинарный режимы открытия файлов, дескрипторно- и потоково-ориентированный ввод-вывод.</p> <p>Лабораторные занятия Функции, вызов, передача параметров, базовые алгоритмы вычисления конечных сумм, рядов. Пользовательские типы данных, структуры, объединения</p>	2	4 4		Лекционная презентация №1- 4	[1], [4]	Защита отчетов по лаб. работам

1.5.	<p>Алгоритмы и структуры данных.</p> <p>Лекции Понятие алгоритма. Базовые алгоритмы для обработки данных простых типов и структур данных: работа с массивами, сортировка и поиск. Вычислительная сложность алгоритмов. Рекурсивные алгоритмы, примеры рекурсивных алгоритмов. Динамически создаваемые объекты. Списки, очереди, стеки, деревья, строки.</p> <p>Лабораторное занятие Работа с одномерными и двумерными массивами, алгоритмы сортировки и поиска. Работа со списками, стеками, очередями.</p>	2			Лекционная презентация №1- 5	[1], [5]	Защита отчетов по лаб. работам
1.6.	<p>Разработка консольных приложений в ОС Windows и Linux.</p> <p>Лекции Этапы жизненного цикла программ. Разработка и отладка консольных C++ приложений в системе Microsoft Visual Studio. Разработка и отладка консольных приложений под ОС Linux с использованием среды разработки KDevelop.</p> <p>Лабораторное занятие Решение прикладных задач кинематики и динамики. Разработка в среде KDevelop в системе Linux, многофайловый проект.</p>	2			Лекционная презентация № 1-6	[1], [5]	Защита отчетов по лаб. работам
2 семестр		26	24				
2.1.	<p>Основы объектно-ориентированного программирования.</p> <p>Лекции Классы и экземпляры, поля и методы, наследование. Полиморфизм и полиморфные функции. Модификаторы доступа к полям и методам. Статические поля и методы. Конструкторы и деструкторы. Конструкторы по умолчанию и копирования. Абстрактные классы. Виртуальные функции. Перегрузка операций, дружественные функции и классы. Множественное наследование, виртуальные базовые классы.</p> <p>Лабораторное занятие Наследование.</p>	2	2		Лекционная презентация № 2-1	[1], [4], [5]	Защита отчета по лаб. работе

2.2.	<p>Стандартная библиотека C++ и STL. Лекции Шаблоны типов, функций классов. Контейнер string. Работа со строками, инициализация, присваивание, копирование, проблемы. Контейнеры vector, list. Использование vector контейнера для работы с динамическим массивами. Ассоциативные контейнеры map, multimap, hash_map. Итераторы. Специальные контейнеры. Функторы. Интернационализация. Лабораторное занятие Полиморфизм и виртуальные функции</p>	2	2	2	Лекционная презентация № 2-2	[1], [5]	Защита отчета по лаб. работе
2.3.	<p>C++: ввод-вывод, работа с файлами. Лекции Потоки. Ввод-вывод встроенных типов, состояния потока, форматирование, манипуляторы. Ввод-вывод пользовательских типов данных, перегрузка операций ввода-вывода. Лабораторное занятие Конструкторы и деструкторы</p>	2	2	4	Лекционная презентация № 2-3	[1], [5]	Защита отчета по лаб. работе
2.4.	<p>Обработка исключительных ситуаций в C++. Лекции Традиционный подход к обработке ошибок. Концепция исключения, перехват исключения, вложенная обработка исключений. Возбуждение исключения. Стандартные типы исключений, исключительные ситуации при работе с числами. Лабораторные занятия Перегрузка операций Работа с STL</p>	2	2	4 4	Лекционная презентация № 2-4	[1], [4], [5]	Защита отчетов по лаб. работам
2.5.	<p>Разработка приложений в ОС Windows и Linux. Лекции Этапы жизненного цикла программ. Разработка оконных C++ приложений в системе Microsoft Visual Studio. Разработка оконных приложений под ОС Linux. Лабораторное занятие Ввод-вывод и обработка исключительных ситуаций</p>	2	2	4	Лекционная презентация № 2-5	[1], [2], [5]	Защита отчета по лаб. работе

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

Основная:

1. Маклафлин, Б. Объектно-ориентированный анализ и проектирование = Head First Object-Oriented Analysis and Design / Б. Маклафлин, Г. Поллайс, Д. Уэст; [перевел с английского Е. Матвеев]. - Санкт-Петербург [и др.]: Питер, 2018. - 601 с.
2. Свердлов, С. З. Языки программирования и методы трансляции: учебное пособие для вузов / С. З. Свердлов. — 2-е изд., стер. — Санкт-Петербург: Лань, 2019. — 564 с. — Режим доступа: по подписке: URL: <https://e.lanbook.com/book/173116>
3. Юрьева, А. А. Математическое программирование [Электронный ресурс]: учебное пособие / А. А. Юрьева. — 2-е изд., испр. и доп. — Санкт-Петербург: Лань, 2021. — 432 с. // Лань: электронно-библиотечная система. — Режим доступа: по подписке URL: <https://e.lanbook.com/book/168878>
4. Городняя, Л. В. Парадигма программирования [Электронный ресурс]: учебное пособие для вузов / Л. В. Городняя. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 232 с. // Лань: электронно-библиотечная система. — Режим доступа: по подписке: URL: <https://e.lanbook.com/book/151660>

Дополнительная

1. Григорьев, А. Программирование на языке С для персональных компьютеров. Выпуск 9: Мышь. - Москва: Диалог-МИФИ, 1992. - 110с. - (С для РС).
2. Коваль, Г.И. Программирование в системе виртуальных машин ЕС ЭВМ: Справочное издание - Москва: Финансы и статистика, 1990. — 254 с.
3. Кушниренко, А.Г. Программирование для математиков: Учебное пособие для вузов по специальности "Математика" и "Прикладная математика". - Москва: Наука, 1988. — 382 с.

Владимир Туркова Е. В.

ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ РАБОТ

1 семестр

1. Работа в среде Microsoft Visual Studio, разработка консольных приложений.
2. Оператор присваивания, арифметико-логические операции со встроенными типами данных. Элементарные функции ввода-вывода.
3. Работа с основными конструкциями управления.
4. Работа с одномерными и двумерными массивами, алгоритмы сортировки и поиска.
5. Функции, вызов, передача параметров, базовые алгоритмы вычисления конечных сумм, рядов.
6. Пользовательские типы данных, структуры, объединения.
7. Работа со списками, стеками, очередями.
8. Решение прикладных задач кинематики и динамики.
9. Разработка в среде KDevelop в системе Linux, многофайловый проект.

2 семестр

1. Наследование.
2. Полиморфизм и виртуальные функции
3. Конструкторы и деструкторы
4. Перегрузка операций
5. Работа с STL
6. Ввод-вывод и обработка исключительных ситуаций.

ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЁТУ

1 семестр

1. Среда разработки Microsoft Visual Studio. Основные возможности. Примеры применения.
2. Среда разработки Microsoft Visual Studio. Средства отладки. Средства диагностики языка C++. Примеры применения.
3. Структура программы на C++. Пример простой программы.
4. Директивы препроцессора `#define`, `#undef`. Макроопределения препроцессора (с параметрами и без). Условная компиляция `#if`, `#ifdef`, `#ifndef`, `#else`, `#endif`. Примеры применения.
5. Элементы языка C++: алфавит, лексемы языка, идентификатор, ключевые слова, константы, разделители, выражения. Примеры применения.
6. Встроенные типы данных языка C++ (целый тип, числа с плавающей точкой, символьный тип, тип `bool` и другие). Примеры применения.
7. Переменные и константы языка C++. Глобальные переменные. Области действия переменных и констант. Пространство имен. Примеры применения.
8. Правила определения переменных и типов. Инициализация данных. Примеры применения.
9. Время жизни и область видимости программных объектов. Классы памяти.
10. Инициализация глобальных и локальных переменных. Примеры применения.
11. Операции языка C++ (операция присвоения в полной и короткой форме, операции сложения, вычитания, изменения знака, умножения, деления, деление по модулю, операции увеличения и уменьшения). Примеры применения.
12. Структуры и объединения. Поля битов. Доступ к полям структуры. Указатели на структуры. Примеры применения.
13. Перечисляемый тип данных языка C++. Примеры применения.
14. Указатели языка C++. Операции разыменованье и получения адреса. Адресная арифметика. Инициализация указателя. Указатель на тип `void`. Примеры применения.
15. Выражения. Операнды и операции (унарные, бинарные, тернарные). Правила преобразования типов. Примеры применения.
16. Одномерные и многомерные массивы в языке C++. Индексные выражения. Хранение в памяти одномерных и многомерных массивов. Примеры применения.

17. Объявление, инициализация массивов, обращение к элементам массива в языке C++. Связь между указателями и массивами. Примеры применения.
18. Динамическое распределение памяти. Библиотечные функции для выделения и освобождения динамической памяти. Оператор `sizeof()`. Примеры применения.
19. Динамические массивы в языке C++. Особенности выделения и освобождения памяти для многомерных массивов. Использование операторов `new` и `delete`. Примеры применения.
20. Основные алгоритмы обработки массивов в языке C++. Ввод-вывод, поиск экстремума, сортировка. Примеры применения.
21. Условные операции: меньше (больше), меньше (больше) или равно, равенство, неравенство, логическое «И» и «ИЛИ» в языке C++. Примеры применения.
22. Составной оператор. Операторы ветвления `if`, `if/else`, `switch/case` в языке C++. Примеры применения.
23. Оператор цикла `for` в языке C++. Примеры применения.
24. Оператор цикла с условием `while` в языке C++. Примеры применения.
25. Оператор цикла с постусловием `do/while` в языке C++. Примеры применения.
26. Операторы перехода `break`, `continue`, `return`, `goto` в языке C++. Примеры применения.
27. Определение и вызов функций в языке C++. Фактические и формальные параметры, возвращаемые значения. Примеры применения.
28. Передача массивов и указателей на функции в языке C++. Модификатор `const`. Передача параметров через указатель. Примеры применения.
29. Предварительная инициализация параметров, функции с переменным числом параметров в языке C++. Примеры применения.
30. Передача параметров функции `main` в языке C++. Примеры применения.

ПЕРЕЧЕНЬ ЭКЗАМЕНАЦИОННЫХ ВОПРОСОВ

2 семестр

1. Функции стандартной библиотеки языка C++. Объявление, определение, параметры, возвращаемые значения. Достоинства и недостатки. Примеры применения.
2. Рекурсивные функции в языке C++. Объявление, определение, параметры,
3. возвращаемые значения. Достоинства и недостатки. Примеры применения.
4. Строки в языке C++. Объявление и инициализация массива символов (строк). Нулевой символ. Библиотечные функции работы со строками. Примеры применения.
5. Строки в языке C++. Создание строк. Конструкторы и деструктор строк. Примеры применения.
6. Операции над строками в языке C++. Присваивание и добавление частей строк. Примеры применения.
7. Функции поиска и сравнения подстрок в языке C++. Примеры применения.
8. Функции преобразования и получения характеристик строк в языке C++. Примеры применения.
9. Классы в языке C++. Инкапсуляция. Описание класса. Рекомендации по составу класса.
10. Конструкторы и деструкторы классов в языке C++.
11. Структуры, объединения и классы в языке C++. Общее и отличия. Примеры применения.
12. Конструктор копирования в языке C++. Поверхностное и глубинное копирование. Конструкторы и присваивание строк. Примеры применения.
13. Указатели на элементы классов в языке C++. Указатель this.
14. Константные поля и методы класса в языке C++. Примеры применения.
15. Статические элементы класса в языке C++. Статические поля и статические методы. Примеры применения.
16. Полиморфизм в языке C++. Виртуальные методы классов. Примеры применения.
17. Иерархия наследования классов в языке C++. Доступ к членам базовых классов. Ключи доступа. Примеры применения.

18. Простое и множественное наследование в языке C++. Примеры применения.
19. Виртуальные базовые классы в языке C++. Примеры применения.
20. Виртуальные деструкторы в языке C++. Примеры применения.
21. Друзья класса в языке C++. Дружественные функции и поля. Дружественный класс. Примеры применения.
22. Механизм позднего связывания в языке C++. Примеры применения.
23. Абстрактные классы в языке C++. Абстрактные методы и классы. Примеры применения.
24. Шаблоны функций в языке C++. Объявление, определение, параметры, возвращаемые значения. Достоинства и недостатки. Примеры применения.
25. Шаблоны структур и объединений в языке C++. Определение и инициализация структур-переменных. Примеры применения.
26. Шаблоны классов в языке C++. Создание и использование. Специализация шаблонов классов. Достоинства и недостатки шаблонов классов. Примеры применения.
27. Перегрузка унарных и бинарных операций в языке C++. Примеры применения.
28. Перегрузка операций присваивания в языке C++. Примеры применения.
29. Перегрузка операций new и delete в языке C++. Примеры применения.
30. Перегрузка операции вызова функции в языке C++. Примеры применения.
31. Перегрузка операции индексирования в языке C++. Примеры применения.
32. Перегрузка операций приведения типа в языке C++. Примеры применения.
33. Динамическое определение типа и преобразование типов. Повышающие и понижающие преобразования. Примеры применения.
34. Преобразование ссылок в языке C++. Перекрестное преобразование. Примеры применения.
35. Преобразование типов в языке C++: операции const_cast, static_cast, dynamic_cast, reinterpret_cast. Примеры применения.
36. Обработка исключительных ситуаций в языке C++. Иерархии исключений. Общий механизм обработки исключений. Примеры применения.
37. Синтаксис исключений в языке C++. Список исключений функции. Примеры применения.

38. Исключения в конструкторах и деструкторах языка C++. Перехват исключений. Примеры применения.
39. Поточковые классы в языке C++. Стандартные потоки. Форматирование данных. Флаги и форматирующие методы манипуляторы. Примеры применения.
40. Строковые потоки в языке C++. Примеры применения.
41. Ввод-вывод встроенных (стандартных) типов в языке C++. Примеры применения.
42. Состояния предопределенных объектов (потоков) в языке C++. Ошибки потоков. Примеры применения.
43. Потоки и типы, определенные пользователем в языке C++. Ввод-вывод типов, определенных пользователем. Примеры применения.
44. Форматированный ввод-вывод в языке C++. Манипуляторы. Примеры применения.
45. Файловый ввод-вывод в языке C++. Файловые потоки. Примеры применения.
46. Контейнерные классы в языке C++. Примеры применения.
47. Последовательные контейнеры в языке C++. Вектор (vector). Вектор логических значений vector<bool>. Примеры применения.
48. Последовательные контейнеры в языке C++. Двусторонняя очередь (deque). Примеры применения.
49. Последовательные контейнеры в языке C++. Список (list). Примеры применения.
50. Адаптеры последовательных контейнеров в языке C++. Стек (stack). Примеры применения.
51. Адаптеры последовательных контейнеров в языке C++. Очередь FIFO (queue). Примеры применения.
52. Адаптеры последовательных контейнеров в языке C++. Очередь с приоритетами (priority_queue). Примеры применения.
53. Ассоциативные контейнеры в языке C++. Словари (map), словари с дубликатами (multimap). Примеры применения.
54. Ассоциативные контейнеры в языке C++. Множества (set), множества с дубликатами (multiset), битовые множества (bitset). Примеры применения.
55. Итераторы в языке C++. Обратные итераторы. Итераторы вставки. Поточковые итераторы. Примеры применения.
56. Адаптеры указателей на функции языка C++. Адаптеры методов. Примеры применения.
57. Модифицирующие операции с последовательностями в языке C++. Примеры применения.

58. Средства для численных расчетов и работы с комплексными числами в языке C++. Примеры применения.
59. Средства поддержки языка и локализации. Примеры применения.
60. Текстовые и бинарные файлы в языке C++. Связывание файловых переменных с внешней средой. Примеры применения.
61. Типовые действия с файлами в языке C++: создание, открытие, закрытие, чтение и изменение. Примеры применения.
62. Последовательный и произвольный доступ к файлу в языке C++.
Примеры применения.

СРЕДСТВА ДИАГНОСТИКИ РЕЗУЛЬТАТОВ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ

1. Защита письменных отчетов по лабораторным работам.
2. Индивидуальное домашнее задание для внеаудиторного контроля самостоятельной работы в семестре.
3. Краткие сообщения по дополнительным заданиям для углубленного изучения отдельных тем дисциплины.
4. Зачёт
5. Экзамен

КОНТРОЛЬ КАЧЕСТВА УСВОЕНИЯ ЗНАНИЙ

Диагностика качества усвоения знаний проводится в форме промежуточного контроля и текущей аттестации.

Результат промежуточного контроля (Π) за первый и второй семестр оценивается отметкой в баллах по десятибалльной шкале и выводится исходя из отметок, выставленных в ходе проведения мероприятий промежуточного контроля в течение семестра по формуле:

$$\Pi = \frac{\Pi K_1 + \Pi K_2 + \Pi K_3}{3},$$

где каждое из слагаемых определяется как среднеарифметическое по каждой из форм промежуточного контроля.

Допускается по представлению преподавателя и по решению кафедры физики учреждения образования «Полоцкий государственный университет» увеличение результата промежуточного контроля для студентов, принимавших участие в олимпиадах, конкурсах студенческих работ и т.д. (согласно приказу №294 от 06.06.2014)

Текущая аттестация по дисциплине в первом семестре проводится в форме зачёта.

Зачёт проводится в соответствии с Правилами проведения аттестации студентов, курсантов, слушателей при освоении содержания образовательных программ высшего образования, утвержденными постановлением Министерства образования Республики Беларусь 29.05.2012 № 53.

Отметка по зачёту (O) рассчитывается на основе отметки по десятибалльной шкале, полученной студентом за ответы на вопросы по билету.

Итоговая экзаменационная отметка по дисциплине ($З$) рассчитывается на основе результата промежуточного контроля (Π) и отметки, полученной студентом за ответ по билету (O) по формуле

$$k\Pi + (1 - k)O = З$$

где k – весовой коэффициент промежуточного контроля (утверждается в начале семестра на заседании кафедры и доводится до сведения студентов на первом занятии).

Текущая аттестация по дисциплине во втором семестре проводится в форме экзамена.

Экзамен проводится в соответствии с Правилами проведения аттестации студентов, курсантов, слушателей при освоении содержания образовательных программ высшего образования, утвержденными постановлением Министерства образования Республики Беларусь 29.05.2012 № 53 и Положением о рейтинговой системе оценки знаний и компетенций студентов (приказ № 294 от 06.06. 2014г.).

Отметка по экзамену (О) рассчитывается на основе отметки по десятибалльной шкале, полученной студентом за ответы на вопросы по билету.

Итоговая экзаменационная отметка по дисциплине (Э) рассчитывается на основе результата промежуточного контроля (П) и отметки, полученной студентом за ответ по билету (О) по формуле

$$k\Pi + (1 - k)O = \text{Э}$$

где k – весовой коэффициент промежуточного контроля (утверждается в начале семестра на заседании кафедры и доводится до сведения студентов на первом занятии).

ХАРАКТЕРИСТИКА ИННОВАЦИОННЫХ ПОДХОДОВ К ПРЕПОДАВАНИЮ УЧЕБНОЙ ДИСЦИПЛИНЫ

При изучении дисциплины «Программирование» используются следующие инновационные подходы:

- Проведение всех видов аудиторных занятий осуществляется с применением компьютерных технологий.

- Проведение лабораторного практикума по дисциплине основано на внедрении поисково-проектного метода (для выполнения заданий лабораторной работы предоставляется индивидуальное рабочее место, укомплектованное персональным компьютером с установленным программным обеспечением, для подготовки и оформления письменного отчета, углубленного изучения темы при подготовке к защите результатов, полученных в работе).

- Для оценки знаний и компетенций студентов используется рейтинговая система (дневная форма получения образования).

- Контроль качества усвоения знаний осуществляется на основе индивидуального дифференцированного подхода при составлении заданий внеаудиторной контрольной работы для самоконтроля. Отчет о выполнении заданий представляется, согласно учебно-методической карте дисциплины на 17 неделе текущего семестра обучения. Работа над заданием является индивидуальной, роль преподавателя в ней – консультационная в течение семестра.

- Отдельные темы учебной дисциплины «Программирование» предлагаются обучаемым для углубленного изучения, что позволяет применять поисковый метод в образовательном процессе. Проект при этом носит индивидуальный характер и направление на формирование навыков самостоятельного поиска информации, умения ее анализировать и систематизировать, что позволяет выработать у обучающегося базовые профессиональные компетенции, направленные на получение навыков самообразования, а также поиска, систематизации и представления технической информации. Формы отчетности по указанному виду работы: конспект, краткое письменное сообщение, компьютерная презентация, доклад на научной студенческой конференции.

- материалы для подготовки ко всем видам проводимых занятий выкладываются на базе платформы Google Classroom.

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении дисциплины используются следующие формы самостоятельной работы:

- углубленное изучение теоретического материала при подготовке к лабораторным работам;
- подготовка письменных отчетов по выполненным лабораторным работам;
- самостоятельная работа в виде выполнения внеаудиторной индивидуальной домашней работы;
- подготовка кратких сообщений по дополнительным заданиям для углубленного изучения отдельных тем дисциплины.

Содержание самостоятельной работы студентов

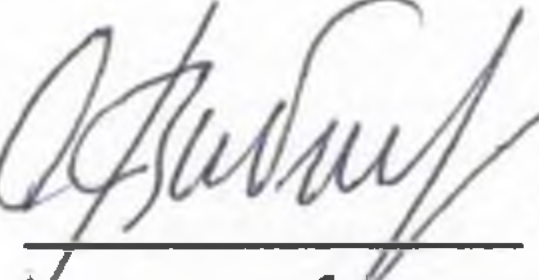
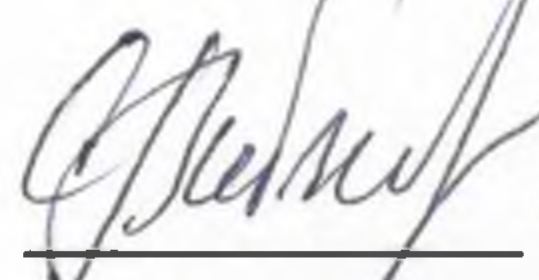
Вид самостоятельной работы	Тематическое содержание и используемые источники	Количество часов
1 семестр		
Оформление письменного отчета по лабораторной работе и подготовка к защите лабораторных работ	Лабораторная работа №1*	2
	Лабораторная работа №2*	2
	Лабораторная работа №3*	2
	Лабораторная работа №4*	2
	Лабораторная работа №5*	2
	Лабораторная работа №6*	2
	Лабораторная работа №7*	2
	Лабораторная работа №8*	2
	Лабораторная работа №9*	2
Выполнение индивидуальной внеаудиторной домашней работы	Осн. литература: [1, 2, 4-7], конспект лекций Доп. литература: [1-6]	12
Подготовка краткого сообщения по отдельным углубленно изучаемым темам**	Осн. литература: [1, 2, 4-7], конспект лекций Доп. литература: [1-6] Интернет-ресурсы открытого доступа	10
Подготовка к зачёту	Перечень вопросов к зачёту. Осн. литература: [2-4, 6, 7], конспект лекций Доп. литература: [1-6]	14
	Всего часов за 1 семестр:	54
2 семестр		
Оформление письменного отчета по лабораторной работе и подготовка к защите лабораторных работ	Лабораторная работа №1*	2
	Лабораторная работа №2*	2
	Лабораторная работа №3*	2
	Лабораторная работа №4*	2
	Лабораторная работа №5*	2
	Лабораторная работа №6*	2
Выполнение индивидуальной внеаудиторной домашней работы	Осн. литература: [1, 2, 4-7], конспект лекций Доп. литература: [1-6]	12

Подготовка краткого сообщения по отдельным углубленно изучаемым темам**	Осн. литература: [1, 2, 4-7], конспект лекций Доп. литература: [1-6] Интернет-ресурсы открытого доступа	10
Подготовка к экзамену	Перечень экзаменационных вопросов. Осн. литература: [2-4, 6, 7], конспект лекций Доп. литература: [1-6]	24
	Всего часов за 2 семестр:	58
	ИТОГО:	112

* - тема работы соответствует индивидуальному графику обучающегося, составленному в начале семестра из расчета общего количества часов, отводимых в соответствии с учебной программой.

** - право выбора темы для углубленного изучения предоставляется студентам в соответствии с учебно-методической картой дисциплины.

**ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ
ПО ИЗУЧАЕМОЙ ДИСЦИПЛИНЕ
С ДРУГИМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ**

Название дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Программно-аппаратные интерфейсы информационных систем	Кафедра физики	<p><i>Замечаний и предложений нет</i></p> <p> С.А. Вабищевич зав. кафедрой физики</p>	
Объектно-ориентированное программирование		<p><i>Замечаний и предложений нет</i></p> <p> С.А. Вабищевич зав. кафедрой физики</p>	

РЕЦЕНЗИЯ

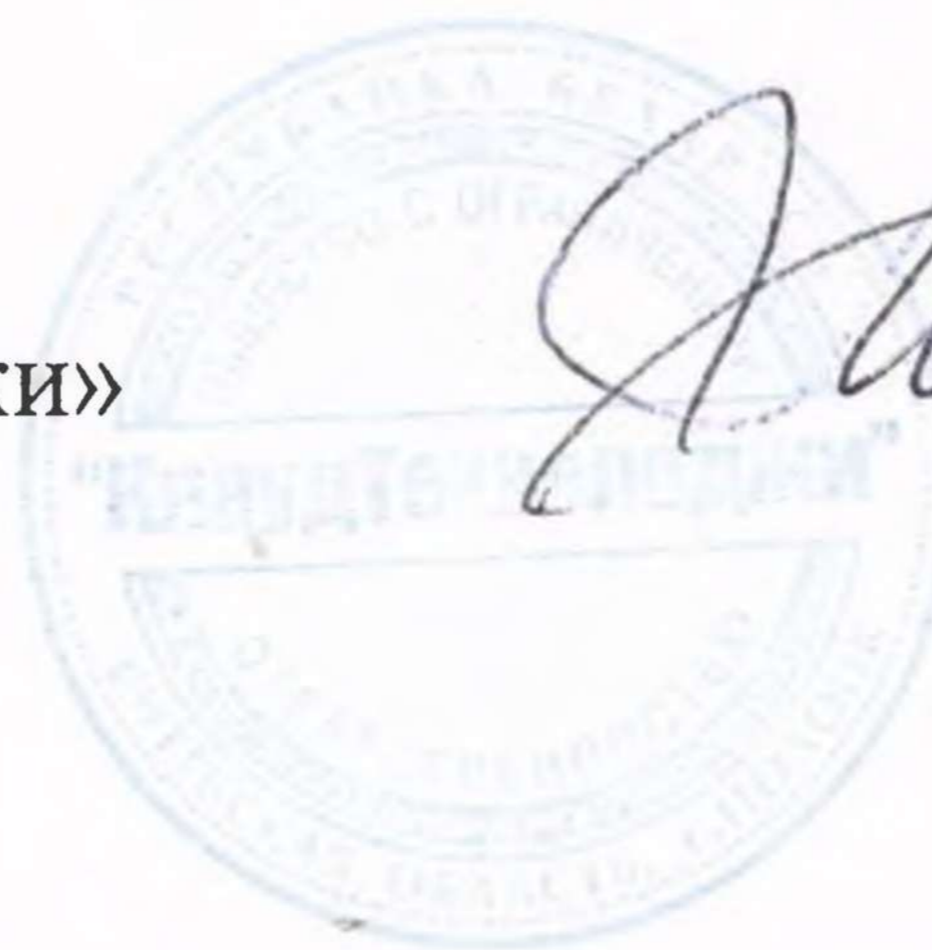
на учебную программу учреждения высшего образования по учебной дисциплине «Программирование»
для специальности 1-31 04 08 «Компьютерная физика»
составленную заведующим кафедрой физики, к.ф.-м.н Вабищевич С.А. и старшим преподавателем кафедры физики Солдатенко П. Н.

Рассмотренная учебная программа учреждения высшего образования по учебной дисциплине «Программирование» предназначена для подготовки специалистов с высшим образованием по специальности 1-31 04 08 «Компьютерная физика» и нацелена на приобретение будущими специалистами в области компьютерного моделирования физических процессов теоретических и практических навыков применения современных методов прикладного программирования. Преподавание дисциплины «Программирование» ставит целями освоение студентами основных конструкций языков программирования С и С++, основных структур данных и алгоритмов их обработки, а также получение начальных навыков разработки программного обеспечения, работающего под управлением операционных систем Windows и Linux. Содержание учебного материала структурировано и направлено на изучение основ алгоритмизации и программирования последовательно, начиная с изучения структуры и принципа работы ЭВМ и его основных компонентов, переходя к основным конструкциям языков С и С++, что позволяет начинать изучение дисциплины, не имея навыков программирования. Практическая часть рассмотренной программы представлена в виде лабораторного практикума. Темы лабораторных работ предоставляют возможность практического изучения конструкций С++ и основных алгоритмов работы с данными. Немаловажную роль в подготовке специалистов в области компьютерного моделирования физических процессов является применение навыков программирования в процессе решения физических задач. В лабораторном практикуме предусмотрено алгоритмическое решение прикладных задач кинематики и динамики.

Полученные в процессе изучения дисциплины, знания и навыки являются базой для дальнейшего изучения дисциплины «Программно-аппаратные интерфейсы информационных систем» и других.

Учебная программа может быть рекомендована к утверждению в качестве дисциплины, входящей в «Модуль программирование» государственного компонента типового учебного плана подготовки специалистов с высшим образованием по специальности 1-31 04 08 «Компьютерная физика».

Генеральный директор
ООО «КлаудТехнолоджи»



Т. М. Глухова

РЕЦЕНЗИЯ

на учебную программу учреждения высшего образования по учебной дисциплине

«Программирование»

для специальности 1-31 04 08 «Компьютерная физика»

составленную заведующим кафедрой физики, к.ф.-м.н Вабищевич С.А. и старшим преподавателем кафедры физики Солдатенко П. Н.

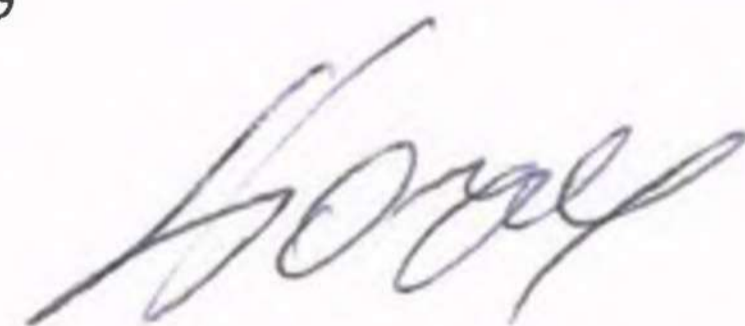
Разработанная учебная программа учреждения высшего образования по учебной дисциплине «Программирование» предназначена для подготовки специалистов с высшим образованием по специальности 1-31 04 08 «Компьютерная физика» и нацелена на формирование у будущих специалистов навыков применения в своей профессиональной деятельности современных методов программирования. Изучение программного материала направлено на формирование базовых понятий структурного программирования, изучении основ алгоритмизации задачи и поиска пути её решения, обучении элементарным алгоритмам обработки структурной информации и т.д. В рассмотренной программе изучаемым языком программирования выбран компилируемый, строго типизированный язык общего назначения C++. Содержание учебного материала основано на последовательном изучении аппаратных и программных средств ЭВМ, сравнительном анализе алгоритмических языков программирования и изучении основных конструкций, функций и библиотек C и C++. На изучение дисциплины отводится 104 аудиторных часа, из них 60 часов лабораторных занятий. Представленный перечень тем лабораторных работ позволяет в полном объёме получить не только теоретические, но и практические навыки работы с основными конструкциями и алгоритмами, а также применить полученные навыки алгоритмизации при решении прикладных задач физики.

Учебная программа соответствует всем требованиям высшей школы. и может быть рекомендована к утверждению в качестве дисциплины, входящей в «Модуль программирование» государственного компонента типового учебного плана подготовки специалистов с высшим образованием по специальности 1-31 04 08 «Компьютерная физика».

Заведующий кафедрой

вычислительных систем и сетей,

к.т.н., доцент



Р. П. Богуш

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УВО

на 2022 / 2023 учебный год

№ п/п	Дополнения и изменения	Основание
1.	Дополнений и изменений нет	

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ физики _____ (протокол № 1 от 31.08.2022 г.)

Заведующий кафедрой физики
к.ф.-м.н., доцент



Вабищевич С.А.

УТВЕРЖДАЮ
Декан ФКНЭ
к.э.н.



Галешова Е.И.