

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»
ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра Компьютерных технологий

УТВЕРЖДАЮ:

проректор по научно-методической
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ПРОГРАММИРОВАНИЕ»

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Профиль подготовки: **Информатика и вычислительная техника**

Образовательная программа: **бакалавриат**

Квалификация: **академический бакалавр**

Форма обучения: **очная, очно-заочная, заочная, в том числе с ускоренным сроком обучения**

Донецк 2020



УТВЕРЖДАЮ:

Декан физико-технического факультета

Фоменко С.А.

«17» апреля 2020 г.

М.П.

Программа учебной дисциплины «Программирование» составлена на основе Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённого приказом Министерства образования и науки ДНР от «21» января 2016 г. №31»; «Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики», утверждённого приказом Министерства образования и науки ДНР №1171 от «10» ноября 2017 г.»; учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 09.03.01 Информатика и вычислительная техника, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Доцент кафедры компьютерных технологий

Шарий Т.В.

Программа учебной дисциплины утверждена на заседании кафедры компьютерных технологий

Протокол № 12 от «2» апреля 2020 г.

Зав. кафедрой компьютерных технологий

Ермоленко Т.В.

Программа учебной дисциплины одобрена учебно-методической комиссией физико-технического факультета

Протокол № 5 от «15» апреля 2020 г.

Председатель учебно-методической комиссии физико-технического факультета

Котенко В.Н.

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Программирование» относится к вариативной части профессионального блока и состоит из четырех содержательных модулей: модуль 1 – «Основы объектно-ориентированного программирования», модуль 2 – «Дополнительные главы объектно-ориентированного программирования», модуль 3 – «Основы языка Python», модуль 4 – «Скриптовое программирование на языке Python».

Основывается на базе дисциплин: «Основы программирования», «Информатика и информационно-коммуникационные технологии». Является основой для изучения следующих дисциплин: «WEB-программирование», «Объектно-ориентированное программирование», «Тестирование и внедрение программного обеспечения».

2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>				
Направление подготовки	09.03.01 Информатика и вычислительная техника			
Профиль	Информатика и вычислительная техника			
Образовательная программа	Бакалавриат			
Квалификация	Академический бакалавр			
Количество содержательных модулей	4			
Дисциплина базовой / вариативной части образовательной программы	Профессиональный блок. Вариативная часть			
Формы контроля (МК, экзамен, зачет)	Два модульных контроля, два экзамена			
Показатели	очная форма обучения		заочная форма обучения	
	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачётных единиц (кредитов)	8	8	8	8
Год подготовки	2	1	2	1
Семестр	3, 4	1, 2	3, 4	1, 2
Количество часов	288(180,108)	288(180,108)	288 (180,108)	288 (180,108)
- лекционных	68 (36,32)	68 (36,32)	16 (8, 8)	16 (8, 8)
- практических, семинарских				
- лабораторных	68 (36,32)	68 (36,32)	16 (8, 8)	16 (8, 8)
- самостоятельной работы	152 (108,44)	152(108,44)	256 (164,92)	256 (164,92)
в т. ч. индивидуальное задание				
Недельное количество часов, т. ч.	10 (10,10)	10 (10,10)	10 (10,10)	10 (10,10)
аудиторных	4 (4,4)	4 (4,4)	1 (1,1)	1 (1,1)

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цель – формирование у студентов базовых знаний и навыков, связанных с программированием объектно-ориентированных, системных, графических, а также научно-исследовательских приложений на языках высокого уровня C++ и Python.

Задачи – изучение основ объектно-ориентированного и обобщенного программирования на примере языка C++; изучение основ функционального и объектно-ориентированного программирования на скриптовом языке Python, возможностей Python для работы с файловой системой и проведения научных исследований.

Требования к результатам освоения дисциплины:

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО по данному направлению подготовки (профилю):

а) общекультурных (ОК):

способность работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия (ОК-6);

способность к самоорганизации и самообразованию (ОК-7).

б) общепрофессиональных (ОПК):

основательная подготовка по математике для использования математического аппарата при решении прикладных и научных задач в области компьютерной инженерии (ОПК-1);

знание современных методов построения и анализа алгоритмов, основ численных методов и умение их использовать на практике (ОПК-4).

в) профессиональных (ПК):

проектно-конструкторская деятельность:

пользоваться методиками использования программных средств для решения практических задач (ПК-2);

использовать и самостоятельно разрабатывать интерфейсы взаимодействия человека и ЭВМ (ПК-3);

знание принципов программирования, средств современных языков программирования, структур данных (ПК-5);

подготовка компьютерных систем с параллельной или распределенной архитектурой; владение современными языками и библиотеками параллельного программирования (ПК-8);

знание методологических принципов построения современных компьютерных систем разной организации для высокопродуктивной обработки информации (ПК-12).

Научно-исследовательская деятельность:

базовые знания научно-методических основ и стандартов в области компьютерной инженерии, проводить эксперимент по проверке корректности решений, рассчитывать экономическую эффективность (ПК-15);

умение готовить и проводить доклады с использованием современных компьютерных средств, писать научно-технические отчеты, оформлять результаты исследований в виде статей (ПК-16).

Научно-педагогическая деятельность:

готовить конспекты лекций, проводить повышение квалификации сотрудников (ПК-17).

Сервисно-эксплуатационная деятельность:

инсталлировать, настраивать и сопровождать программное и аппаратное обеспечение информационных и автоматизированных систем (ПК-21).

В результате изучения учебной дисциплины студент должен

Знать:

- основы статической и динамической типизации, особенности управления памятью в C++ и Python;
- разницу между интерпретируемым и компилируемым языком программирования;
- принципы ООП и механизмы повторного использования кода;
- основные структуры данных Python, понятия изменяемых и неизменяемых типов;
- стандарты стиля кода (PEP8, Google code style);
- основы программирования приложений с графическим интерфейсом и элементы системного программирования на языке Python;
- основные возможности научных библиотек Python.

Уметь:

- проектировать и кодировать структуры данных и алгоритмы в рамках объектно-ориентированной и функциональной парадигмы на языках высокого уровня;
- корректно работать с памятью в C++ и типами данных в Python: строками, списками, кортежами, словарями, множествами;
- создавать приложения с оконным интерфейсом;
- писать Python-скрипты для работы с файловой системой и базами данных;
- писать Python-скрипты для научно-исследовательских и прикладных задач.

Владеть:

- навыками разработки программ на языке C++ и скриптов на языке Python для решения широкого круга практических задач.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Порядковый номер и тема	Краткое содержание темы
	<i>Содержательный модуль 1.</i> Основы объектно-ориентированного программирования
Тема 1. Основы языка C++. Работа с памятью	Конструкции и ключевые слова языка C++. Работа с памятью. Массивы. Работа с консолью и файлами.
Тема 2. Классы и объекты	Классы и объекты. Принцип инкапсуляции. Статические члены классов. Константные объекты и методы класса.
Тема 3. Перегрузка операторов	Перегрузка функций. Перегрузка унарных и бинарных операторов. Перегрузка оператора []. Перегрузка префиксного и постфиксного инкремента и декремента.
Тема 4. Наследование	Реализация межклассового отношения типа “is-a”. Переопределение, доопределение и сокрытие методов в подклассе. Модификатор доступа protected. Спецификаторы наследования. Порядок создания и удаления подклассов.
Тема 5. Полиморфизм. Абстрактные классы и интерфейсы	Позднее связывание. Виртуальные функции, таблицы vtable. Виртуальный деструктор и его назначение. Чисто виртуальные функции. Абстрактные классы и интерфейсы.
	<i>Содержательный модуль 2.</i> Дополнительные главы объектно-ориентированного программирования
Тема 6. Агрегация и композиция	Реализация межклассового отношения типа “has-a”. Принцип делегирования.
Тема 7. Дружественные функции и классы	Дружественные функции и классы. Перегрузка операторов с помощью дружественных методов.
Тема 8. Исключения	Концепция исключений как способа передачи управления, альтернативы исключениям. Генерация исключения в методе (throw), перехват исключения (try-catch) Раскручивание стека. Написание собственного исключения. Исключения стандартной библиотеки C++.

Тема 9. Механизмы RTTI (Run-Time Type Identification) в языке C++	Особенности приведения типов и указателей в C++. Оператор <code>dynamic_cast</code> . Структура <code>type_info</code> , функция <code>typeid</code> .
Тема 10. Шаблоны и макросы	Шаблонные функции. Шаблонные классы. Явная и частичная специализация шаблона. Макросы.
Тема 11. Библиотеки STL и boost. Идиомы C++	Контейнеры и итераторы STL. Алгоритмы STL. Функторы. Введение в библиотеку boost. Идиомы RAII и SFINAE. Умные указатели
Тема 12. Стандарты C++11, C++14	Автоматический вывод типа. Лямбда-выражения. Семантика перемещения. Дополнительные ключевые слова и классы стандартной библиотеки C++.
	Содержательный модуль 3. Основы языка Python
Тема 13. Введение в Python	Интерпретатор Python. Виртуальная машина Python. Основы языка, экосистема Python. Базовые конструкции. Обработка исключений в языке Python. Стандарт кода PEP8.
Тема 14. Типы данных Python	Динамическая типизация в языке Python. Изменяемые и неизменяемые типы. Строки, списки, словари, кортежи, множества. Срезы. Регулярные выражения, модуль <code>re</code> .
Тема 15. Элементы функционального программирования в языке Python	Пользовательские функции. Области видимости. Упаковка и распаковка аргументов. Анонимные функции. Функции <code>map</code> , <code>filter</code> , <code>reduce</code> , <code>zip</code> . Модули <code>itertools</code> и <code>functools</code> . Декораторы. Генераторы и итераторы.
Тема 16. Работа с файлами, файловой системой и процессами в Python	Основные возможности Python по работе с файлами и файловой системой. Модули <code>os</code> , <code>shutil</code> , <code>pathlib</code> , <code>glob</code> . Работа с датой и временем. Аргументы командной строки, модуль <code>argparse</code> . Сериализация <code>pickle</code> , <code>json</code> , <code>yaml</code> . Работа с внешними процессами. Модуль <code>subprocess</code> .
	Содержательный модуль 4. Скриптовое программирование на языке Python
Тема 17. Объектно-ориентированное программирование в Python	Особенности инкапсуляции, наследования и полиморфизма в Python. Статические методы и методы класса. Перегрузка операторов, <code>magic methods</code> . Абстрактные классы. Метаклассы и метапрограммирование. Классы исключений. Менеджеры контекста. Глубокое копирование объектов. Модуль <code>copy</code> .
Тема 18. Программирование приложений с графическим интерфейсом пользователя	Программирование приложений с графическим интерфейсом пользователя: <code>wxPython</code> , <code>tkInter</code> , <code>PyQt</code> . Работа с изображениями: <code>Pillow</code> , <code>wxImage</code> . Работа с бинарными данными. Модуль <code>struct</code> .
Тема 19. Работа с базами данных в Python	Работа с реляционными базами данных. Интерфейс на основе <code>cursor</code> . Работа с <code>noSQL</code> базой данных <code>Mongo</code> . Объектно-реляционные отображения (ORM). Библиотека <code>SQLAlchemy</code> .
Тема 20. Многопоточность в Python	Многопоточность и асинхронность в Python. Блокировка GIL. Очереди. Средства синхронизации потоков. Модули <code>threading</code> , <code>asyncio</code> . Элементы сетевого программирования на Python: модули <code>socket</code> , <code>requests</code> , <code>BeautifulSoup</code> .

Тема 21. Язык Python в научных исследованиях	Модули numpy, scipy, sympy, pandas, matplotlib.
--	---

Курс дисциплины «Программирование» предусматривает следующие **формы организации учебного процесса:**

1. лекции;
2. лабораторные занятия;
3. самостоятельная работа студента.

По источнику передачи и восприятия учебной информации используются словесные (лекция, беседа), наглядные (слайды, иллюстрации, коды программ), практические (исследования, упражнения, лабораторные работы) методы.

По характеру познавательной деятельности студентов используются объяснительно-иллюстративные и репродуктивные методы, проблемное преподавание, частично-поисковый и исследовательский методы.

В зависимости от основной дидактической цели и задач используются методы устного изложения знаний, закрепление учебного материала, самостоятельной работы студентов по осмыслению и усвоению нового материала, работы по применению знаний на практике и выработке умений и навыков, проверки и оценки знаний, умений и навыков.

Используются следующие методы контроля:

1. устный контроль (экспресс-опрос на лекциях);
2. проверка конспектов;
3. защита лабораторных работ;
4. проверка самостоятельных работ;
5. модульные контрольные работы;
6. курсовая работа;
7. итоговый контроль (экзаменационные билеты).

Тематический план

	Содержательный модуль 1																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма обучения												Заочная форма обучения										
	Нормативный срок обучения						Ускоренный срок обучения						Нормативный срок обучения						Ускоренный срок обучения				
	всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.			
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные работы	самостоятельная работа
Тема 1. Основы языка C++. Работа с памятью	16	4		4	8		16	4		4	8		16	2		2	12		16	2		2	12
Тема 2. Классы и объекты	20	4		4	12		20	4		4	12		14	1		1	12		14	1		1	12
Тема 3. Перегрузка операторов	20	4		4	12		20	4		4	12		16	1		1	14		16	1		1	14
Тема 4. Наследование	14	2		2	10		14	2		2	10		14	1		1	12		14	1		1	12
Тема 5. Полиморфизм. Абстрактные классы и интерфейсы	20	4		4	12		20	4		4	12		14	1		1	12		14	1		1	12
Итого по 1-му содержательному модулю	90	18		18	54		90	18		18	54		74	6		6	62		74	6		6	62

	Содержательный модуль 2																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма обучения												Заочная форма обучения										
	Нормативный срок обучения						Ускоренный срок обучения						Нормативный срок обучения						Ускоренный срок обучения				
	всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.			
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные работы	самостоятельная работа
Тема 6. Агрегация и композиция	12	2		2	8		12	2		2	8		14				14		14				14
Тема 7. Дружественные функции и классы	12	2		2	8		12	2		2	8		14				14		14				14
Тема 8. Исключения	12	2		2	8		12	2		2	8		16	1		1	14		16	1		1	14
Тема 9. Механизмы RTTI (Run-Time Type Identification) в языке C++	12	2		2	8		12	2		2	8		14				14		14				14
Тема 10. Шаблоны и макросы	16	4		4	8		16	4		4	8		16	1		1	14		16	1		1	14
Тема 11. Библиотеки STL и boost. Идиомы C++	18	4		6	8		18	4		6	8		16				16		16				16
Тема 12. Стандарты C++11, C++14	8	2			6		8	2			6		16				16		16				16
Итого по 2-му содержательному модулю	90	18		18	54		94	18		18	54		106	2		2	102		106	2		2	102

	Содержательный модуль 3																							
Названия содержательных модулей и тем	Количество часов																							
	Очная форма обучения												Заочная форма обучения											
	Нормативный срок обучения						Ускоренный срок обучения						Нормативный срок обучения						Ускоренный срок обучения					
	всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные работы	самостоятельная работа	
Тема 13. Введение в Python	12	4		4	4		12	4		4	4		12	1		1	10		12	1		1	10	
Тема 14. Типы данных Python	14	4		4	6		14	4		4	6		14	1		1	12		14	1		1	12	
Тема 15. Элементы функционального программирования в языке Python	14	4		4	6		14	4		4	6		14	1		1	12		14	1		1	12	
Тема 16. Работа с файлами, файловой системой и процессами в Python	14	4		4	6		14	4		4	6		14	1		1	12		14	1		1	12	
Итого по 3-му содержательному модулю	54	16		16	22		54	16		16	22		54	4		4	46		54	4		4	46	

	Содержательный модуль 4																			
Названия содержательных модулей и тем	Количество часов																			
	Очная форма обучения										Заочная форма обучения									
	Нормативный срок обучения					Ускоренный срок обучения					Нормативный срок обучения					Ускоренный срок обучения				
	всего	В Т. Ч.				всего	В Т. Ч.				всего	В Т. Ч.				всего	В Т. Ч.			
		лекции	практические	лабораторные	самостоятельная работа		лекции	практические	лабораторные	самостоятельная работа		лекции	практические	лабораторные	самостоятельная работа		лекции	практические	лабораторные работы	самостоятельная работа
Тема 17. Объектно-ориентированное программирование в Python	18	4		4	10		18	4		4	10		12	1		1	10		12	1
Тема 18. Программирование приложений с графическим интерфейсом пользователя	14	2		2	10		14	2		2	10		12	1		1	10		12	1
Тема 19. Работа с базами данных в Python	18	4		4	10		18	4		4	10		12	1		1	10		12	1
Тема 20. Многопоточность в Python	20	4		4	12		20	4		4	12		8	1		1	6		8	1
Тема 21. Язык Python в научных исследованиях	16	2		2	12		16	2		2	12		10			10			10	
Итого по 4-му содержательному модулю	86	16		16	22		86	16		16	54		54	4		4	46		54	4
Всего часов	288	68		68	152		288	68		68	152		288	16		16	256		288	16

5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Темы лекционных занятий

№ п/п	Название темы	Количество часов
1.	Основы языка C++. Работа с памятью	4
2.	Классы и объекты	4
3.	Перегрузка операторов	4
4.	Наследование	2
5.	Полиморфизм. Абстрактные классы и интерфейсы	4
6.	Агрегация и композиция	2
7.	Дружественные функции и классы	2
8.	Исключения	2
9.	Механизмы RTTI (Run-Time Type Identification) в языке C++	2
10.	Шаблоны и макросы	4
11.	Библиотеки STL и boost. Идиомы C++	4
12.	Стандарты C++11, C++14	2
13.	Введение в Python	4
14.	Типы данных Python	4
15.	Элементы функционального программирования в языке Python	4
16.	Работа с файлами, файловой системой и процессами в Python	4
17.	Объектно-ориентированное программирование в Python	4
18.	Программирование приложений с графическим интерфейсом пользователя	2
19.	Работа с базами данных в Python	4
20.	Многопоточность в Python	4
21.	Язык Python в научных исследованиях	2
	ВСЕГО	68

Темы лабораторных занятий

№ п/п	Название темы	Количество часов
1.	Работа с памятью в C++.	4
2.	Классы и объекты. Инкапсуляция. Перегрузка операторов.	8
3.	Наследование и полиморфизм. Абстрактные классы и интерфейсы. RTTI.	6
4.	Агрегация и композиция. Дружественные функции и классы. Исключения.	8
5.	Шаблоны. Библиотека STL.	10
6.	Основы языка Python.	12
7.	Работа с файловой системой и процессами в Python.	4
8.	Объектно-ориентированное программирование на языке Python.	6
9.	Прикладные пакеты Python.	10
	ВСЕГО	68

6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Организация самостоятельной работы студентов.

Самостоятельная работа студентов по курсу «Программирование» предусматривает изучение дополнительной технической литературы и интернет-источников, рекомендуемые этой программой; самостоятельную разработку алгоритмов и текстов программ лабораторных работ, изучение дополнительного инструментария. При желании студент может подготовить реферат или доклад по одной из приведенных ниже тем:

1. Сравнительный анализ компиляторов кода C++. Make-файлы.
2. Создание DLL на языке C++.
3. Особенности реализации исключений в языках C++ и Python.
4. Особенности обобщенного программирования. Шаблоны и макросы.
5. Идиомы C++: safe-bool, copy-and-swap, SFINAE, RAII и другие.
6. Принципы написания «безопасного кода» (safe-code).
7. Библиотеки STL, boost, Qt.
8. Стандарты C++11, C++14.
9. Элементы функционального программирования в языке Python.
10. Приложения с графическим интерфейсом пользователя в Python.
11. Работа с реляционными и NoSql-базами данных в Python.
12. Многопоточность в Python. Глобальная блокировка интерпретатора (GIL).
13. Язык Python в научных исследованиях.
14. Тетради IPython/Jupyter Notebook.

№ n/n	Название темы	Количество часов
1.	Основы языка C++. Работа с памятью	8
2.	Классы и объекты	12
3.	Перегрузка операторов	12
4.	Наследование	10
5.	Полиморфизм. Абстрактные классы и интерфейсы	12
6.	Агрегация и композиция	8
7.	Дружественные функции и классы	8
8.	Исключения	8
9.	Механизмы RTTI (Run-Time Type Identification) в языке C++	8
10.	Шаблоны и макросы	8
11.	Библиотеки STL и boost. Идиомы C++	8
12.	Стандарты C++11, C++14	6
13.	Введение в Python	4
14.	Типы данных Python	6
15.	Элементы функционального программирования в языке Python	6
16.	Работа с файлами, файловой системой и процессами в Python	6
17.	Объектно-ориентированное программирование в Python	10
18.	Программирование приложений с графическим интерфейсом пользователя	10
19.	Работа с базами данных в Python	10
20.	Многопоточность в Python	12

21.	Язык Python в научных исследованиях	12
	ВСЕГО	152

7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальные задания предусмотрены к каждой лабораторной работе.
Полный перечень всех заданий приведен в УМКД дисциплины.

Образец индивидуального задания

Написать программу в соответствии с вариантом (из лабораторной работы № 2).

Написать собственный класс в соответствии с вариантом. Продемонстрируйте в коде инкапсуляцию данных, применение конструкторов без параметров и с параметрами для инициализации данных. Класс должен содержать метод *serialize()* для сохранения данных класса в файл и метод *deserialize()* для чтения данных класса из файла по умолчанию в текущей директории, а также перегруженные методы *serialize(const std::string& filename)* и *deserialize(const std::string& filename)* для работы с файлом с указанным в параметре именем.

Вариант 1

Класс «Студент».

Данные: ФИО, пол, год рождения, год поступления, номер зачетки, средний балл.

Создайте трех студентов и установите их личные данные с помощью сеттеров. Еще одного студента создайте отдельно в куче и установите его данные в конструкторе с параметрами. В главной функции симитируйте три сессии: случайным образом сформируйте по четыре новые оценки и пересчитайте в отдельном методе средний балл. Выведите результаты студентов, отсортировав их в порядке убывания среднего балла.

Вариант 2

Класс «Магазин».

Данные: название, адрес, год основания, номер, суммарная прибыль.

Создайте два объекта-магазина в куче и один – в стеке. Данные первых двух заполните с помощью сеттеров, а третий проинициализируйте с помощью конструктора с параметрами. В главной функции симитируйте отдельно продажи за сентябрь, октябрь и ноябрь. Выведите все магазины отдельно в двух рейтинговых списках (сначала первый, затем второй): 1) отсортируйте в порядке убывания общей прибыли за три месяца; 2) отсортируйте в порядке убывания среднего прироста прибыли за три месяца.

Вариант 3

Класс «Музыкант».

Данные: имя, фамилия, пол, год рождения, инструмент, рейтинг.

Создайте трех музыкантов в стеке и одного – в куче. Данные первых трех заполните с помощью сеттеров, а четвертый проинициализируйте с помощью конструктора с параметрами. В главной функции симитируйте три концерта и голосование зрителей, по результатам которого нарастите рейтинг каждого музыканта на случайное число из диапазона 1...5). Выведите список музыкантов в порядке убывания суммарного рейтинга, при этом клавишники должны идти впереди независимо от рейтинга.

...

8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. ООП. Основные принципы. Механизмы повторного использования кода. Структура проекта. Header guards.
2. Классы и объекты. Инициализация объекта класса. Конструкторы и деструкторы. Методы класса, создаваемые компилятором по умолчанию.
3. Классы и объекты. Особенности константных объектов классов. Указатель this. Статические методы класса.
4. Перегрузка операторов. Оператор +, оператор <, оператор [], оператор ++.
5. Принцип инкапсуляции. Спецификаторы доступа. Сеттеры, геттеры. Запрет создания объекта класса. Запрет копирования объекта класса.
6. Наследование. Переопределение, доопределение и сокрытие. Спецификаторы наследования. Порядок создания и удаления подклассов.
7. Полиморфизм. Виртуальные функции.
8. Абстрактные классы и интерфейсы. Чисто виртуальные функции. Виртуальный деструктор.
9. RTTI в C++ (dynamic_cast, typeid). Исключение bad_cast.
10. Композиция и агрегация. Способы реализации на C++. Паттерн делегирования.
11. Дружественные функции и классы. Перегрузка оператора << для потокового вывода произвольного объекта.
12. Исключения в C++. Раскручивание стека.
13. Шаблонные функции и классы. Особенности компиляции шаблонов. Частичная специализация шаблона.
14. Библиотека STL. Контейнеры последовательностей и итераторы.
15. Библиотека STL. Адаптеры контейнеров и итераторы.
16. Библиотека STL. Ассоциативные контейнеры и итераторы.
17. Библиотека STL. Алгоритмы и функторы.
18. Методы класса. Сигнатура, прототип. Перегрузка методов. Перегрузка конструкторов. Списки инициализации.
19. Поточный ввод-вывод в C++. Использование <iostream> и <fstream>. Перегрузка оператора << для потокового вывода произвольного объекта.
20. Работа с памятью в C++. Указатели и ссылки. Типы памяти. Утечки памяти. Массивы.

9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «Донецкий национальный университет»
Физико-технический факультет

Направление подготовки	09.03.01 - Информатика и вычислительная техника
Профиль	Информатика и вычислительная техника
Программа подготовки	бакалавриат
Семестр	3
Учебная дисциплина	Программирование

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №1 ВАРИАНТ №1

1. Классы и объекты. Особенности константных объектов классов. Указатель this. Статические методы класса.
2. Написать класс Circle и код, позволяющий написать следующее:

```
Circle small( 10, 10, 200);
Circle big = small * 3;
```

3. Что будет выведено на консоль в результате выполнения кода:

```
int x[10] = {42};
int* y = x + 3;
int z = *x + *y;
cout << *x << " + " << *y << " = " << z;
```

заменить cout на однострочный printf; направить вывод в файл out.txt.

Утверждено на заседании кафедры компьютерных технологий,
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой
Преподаватель

Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	6
2	6
3	3
<i>Всего</i>	15

ГОУ ВПО «Донецкий национальный университет»

Физико-технический факультет

Направление подготовки 09.03.01 - Информатика и вычислительная техника

Профиль Информатика и вычислительная техника

Программа подготовки бакалавриат

Семестр 4

Учебная дисциплина Программирование

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №2

ВАРИАНТ №1

- Теоретическая часть: генераторы и итераторы.
- Дан список: **words** = ['Hello', 'sos', 'madam', 'is', 'Lena']
На основе списка words получить список кортежей
«Число букв в слове – Флаг того, что слово оканчивается гласным звуком» :
[(5, True), (3, False), (5, False), (2, False), (4, True)]
- На основе списка words получить список всех слов через одно:
['Hello', 'madam', 'Lena']
- С помощью среза получить 2 последних слова в списке words.

Утверждено на заседании кафедры компьютерных технологий,
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой
Преподаватель

Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	5
2	4
3	4
4	2
Всего	15

10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Теоретические вопросы к экзамену (осень)

1. ООП. Основные принципы. Механизмы повторного использования кода. Структура проекта. Header guards.
2. Классы и объекты. Инициализация объекта класса. Конструкторы и деструкторы. Методы класса, создаваемые компилятором по умолчанию.
3. Классы и объекты. Особенности константных объектов классов. Указатель *this*. Статические методы класса.
4. Перегрузка операторов. Оператор +, оператор <, оператор [], оператор инкремента.
5. Принцип инкапсуляции. Спецификаторы доступа. Сеттеры, геттеры. Запрет создания объекта класса. Запрет копирования объекта класса.
6. Наследование. Переопределение, доопределение и сокрытие. Спецификаторы наследования. Порядок создания и удаления подклассов.
7. Полиморфизм. Виртуальные функции.
8. Абстрактные классы и интерфейсы. Чисто виртуальные функции. Виртуальный деструктор.
9. RTTI в C++ (*dynamic_cast*, *typeid*). Исключение *bad_cast*.
10. Композиция и агрегация. Способы реализации на C++. Паттерн делегирования.
11. Дружественные функции и классы. Перегрузка оператора << для потокового вывода произвольного объекта.
12. Исключения в C++. Раскручивание стека.
13. Шаблонные функции и классы. Особенности компиляции шаблонов. Явная и частичная специализация шаблона.
14. Библиотека STL. Контейнеры последовательностей и итераторы.
15. Библиотека STL. Адаптеры контейнеров и итераторы.
16. Библиотека STL. Ассоциативные контейнеры и итераторы.
17. Библиотека STL. Алгоритмы и функторы.
18. Методы класса. Сигнатура, прототип. Перегрузка методов. Перегрузка конструкторов. Списки инициализации.
19. Потоковый ввод-вывод в C++. Использование <iostream> и <fstream>. Перегрузка оператора << для потокового вывода произвольного объекта.
20. Работа с памятью в C++. Указатели и ссылки. Типы памяти. Утечки памяти. Массивы.

ГОУ ВПО «Донецкий национальный университет»
Физико-технический факультет

Направление подготовки 09.03.01 – Информатика и вычислительная техника
Профиль Информатика и вычислительная техника
Программа подготовки бакалавриат
Семестр 3
Учебная дисциплина Программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

1. Классы и объекты. Особенности константных объектов классов. Указатель *this*. Статические методы класса.

2. Напишите интерфейс `Iparser` с методом парсинга (разбора) файлов **`parse(std::string filename)`**. Напишите два класса, реализующие данный интерфейс: `XMLParser` и `MIDIParser`. В первом случае необходимо считать все строки из текстового файла и вывести их количество на консоль; во втором случае – вывести на консоль длину полного имени файла (параметра метода). Если указываемого файла не существует, сгенерируйте исключение. Продемонстрируйте перехват исключения и полиморфизм в функции *main()*.

Утверждено на заседании кафедры компьютерных технологий,
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой _____
Преподаватель _____

Критерии оценивания экзамена

<i>Номер задания</i>	<i>Количество баллов</i>
1	25
2	25
Всего	50

Теоретические вопросы к экзамену (весна)

- Особенности интерпретатора Cpython. Память, переменные, объекты.
- PEP8: стандарт оформления кода на языке Python. Идиомы Python.
- Изменяемые типы данных Python.
- Неизменяемые типы данных Python.
- Генерация и перехват исключений на языке Python.
- Структурирование кода на языке Python. Модули и пакеты.
- Пользовательские функции. Области видимости и правило LEGB.
- Итераторы и генераторы. Модуль `itertools`.
- Декораторы. Модуль `functools`.
- Анонимные функции. Функции `map`, `filter`, `reduce`.
- Работа с файлами и файловой системой в Python.
- Работа с процессами. Модуль `subprocess`.
- Классы и объекты. Свойства, конструкторы, статические члены.
- Перегрузка операторов в классах Python.
- Менеджеры контекста.

16. Скрипты с графическим интерфейсом пользователя в Python.
17. Работа с базами данных в Python.
18. Работа с сетью.
19. Многопоточность в Python. Глобальная блокировка интерпретатора.
20. Пакеты Python для научно-исследовательской деятельности.

ГОУ ВПО «Донецкий национальный университет»
Физико-технический факультет

Направление подготовки 09.03.01 – Информатика и вычислительная техника
Профиль Информатика и вычислительная техника
Программа подготовки бакалавриат
Семестр 4
Учебная дисциплина Программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 7

1. Анонимные функции. Функции map, filter, reduce.
2. Даны два списка по 12 элементов от 1 до 99. Напишите код вывода на консоль третьего списка, в котором сначала идут однозначные числа из первого списка, затем однозначные из второго, затем двузначные элементы из первого, затем двузначные из второго.
3. Напишите код классов «Треклист» и «Песня» (данные: название песни, время звучания); корректно свяжите их. В треклисте перегрузите оператор сложения (он должен выполнять слияние двух треклистов) и оператора in (вхождения песни с определенным названием в треклист). Если при создании песни указывается пустая строка, генерируйте исключение. Продемонстрируйте в коде работу классов и перехват исключения.

Утверждено на заседании кафедры компьютерных технологий,
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой
Преподаватель

Критерии оценивания экзамена

<i>Номер задания</i>	<i>Количество баллов</i>
1	20
2	15
3	15
Всего	50

11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ

Тестовые задания не предусмотрены.

12. КРИТЕРИИ ОЦЕНИВАНИЯ

По курсу предполагается проведение промежуточной аттестации в виде 2 модульных контролей, 2 экзаменов, выполнение лабораторных работ.

*Распределение баллов, которые могут получить студенты
в процессе изучения дисциплины*

Семестр 1

	Содержательный модуль №1				Содержательный модуль №2				Экзамен	Всего
	Лабораторные работы		Мод. контр. работа	Всего С.М. №1	Лабораторные работы			Всего С.М. №2		
	№1	№2			№3	№4	№5			
Макс. балл	7	7	15	29	7	7	7	21	50	100

Семестр 2

	Содержательный модуль №1				Содержательный модуль №2			Экзамен	Всего
	Лабораторные работы		Мод. контр. работа	Всего С.М. №1	Лабораторные работы		Всего С.М. №2		
	№1	№2			№3	№4			
Макс. балл	8	7	15	30	10	10	20	50	100

Шкала соответствия баллов национальной шкале

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (зачет)
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено
C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено
FX	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

Оценка за овладение курса выставляется по следующим критериям:

– Оценку «отлично» заслуживает студент, который обнаружил глубокие знания при ответах на теоретические вопросы по темам курса, а также выполнил практические задания в полном объеме и набрал более 90 баллов.

– Оценку «хорошо» заслуживает студент, сделавший ошибки в теоретических или практических ответах, которые могут быть интерпретированы как малозначительные для вопросов, которые рассматривались. Студент должен набрать более 75 баллов.

– Оценку «удовлетворительно» заслуживает студент, который выполнил задания неполно и с ошибками, но при этом набрал более 60 баллов.

– Оценку «неудовлетворительно» заслуживает студент, который не выполнил большинства теоретических и практических задач и набрал менее 60 баллов.

13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Для проведения лекционных занятий требуется аудитория на поток, оборудованная мультимедийным проектором и экраном, или интерактивной доской. Для проведения лабораторных занятий по дисциплине необходим оборудованный ПЭВМ или ноутбуками компьютерный класс с возможностью выхода в Интернет.

14. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
<i>Основная литература</i>			
1.	Калоеров, С. А. Программирование на языке С++ [Текст] : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. - Донецк : Юго-Восток, 2009. - 298 с.	60	Да
2.	Методические рекомендации к выполнению лабораторных работ по курсу "Программирование" : для студентов направления подготовки 09.03.01 "Информатика и вычислительная техника" / [сост. Т. В. Шарий] ; ГОУВПО "Донецкий национальный университет", Физико-технический факультет, Кафедра компьютерных технологий. - Донецк : ГОУ ВПО "ДонНУ", 2016. - 54 с.	1	Да
<i>Дополнительная литература</i>			
3.	Лафоре Р. Объектно-ориентированное программирование в С++ [Текст] / Р. Лафоре. – СПб.: Питер, 2013. – 928 с.	1	Нет
4.	Мейерс С. Эффективный и современный С++: 42 рекомендации по использованию С++11 и С++14 [Текст] / С. Мейерс. – Вильямс, 2016. – 304 с.	1	Нет
5.	Страуструп Б. Язык программирования С++ [Текст] / Б. Страуструп. – СПб.; М.: Невский диалект; Бином, 2015. – 1136 с.	2	Нет
6.	Саммерфильд М. Программирование на Python 3. Подробное руководство [Текст] / М. Саммерфильд. – М.: Символ-плюс, 2009. – 608 с.	-	Нет

15. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Материалы курса «Программирование на языке С++» (автор – Шарий Т.В.). URL: <https://github.com/ar1st0crat/CppCourse> (дата обращения 18.03.2020 г.).
2. Материалы курса «Программирование на языке Python» (автор – Шарий Т.В.). URL: <https://github.com/ar1st0crat/PythonCourse> (дата обращения 18.03.2020 г.).
3. Документация С++. URL: <http://en.cppreference.com/w/> (дата обращения 18.03.2020).
4. Automate the boring stuff. Practical programming for total beginners. URL: <https://automatetheboringstuff.com/#toc> (дата обращения 18.03.2020).

16. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. IDE Visual Studio Community (версии 2017, 2019).
2. Интерпретатор Python версий от 3.6.
3. Python-дистрибутив Anaconda.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2020 год.
Протокол № 12 от «2» апреля 2020 г.
Заведующий кафедрой Ермоленко Т.В.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2021 год.
Протокол № ____ от «____» _____ 2021 г.
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2022 год.
Протокол № ____ от «____» _____ 2022 г.
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2023 год.
Протокол № ____ от «____» _____ 2023 г.
Заведующий кафедрой