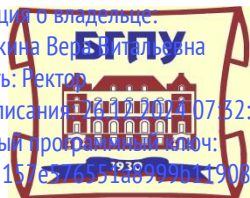


Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Шейкина Версия Татьяна  
Должность: Ректор  
Дата подписания: 2022.06.16 07:32:06  
Уникальный программный ключ:  
a2232a55157e776551a8999b1190892af53989420420336b01573a434e57784



**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования**

**«Благовещенский государственный педагогический университет»**

**ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА**

**Рабочая программа дисциплины**

**«УТВЕРЖДАЮ»**

**Декан**

**Физико-математического факультета**

**ФГБОУ ВО «БГПУ»**

**Т.А. Мерделина**

  
**«16» июня 2022 г.**

**Рабочая программа дисциплины  
ПРОГРАММИРОВАНИЕ**

**Направление подготовки**

**44.03.05 ПЕДАГОГИЧЕСКОЕ ОБРАЗОВАНИЕ  
(с двумя профилями подготовки)**

**Профиль  
«ИНФОРМАТИКА»**

**Профиль  
«ФИЗИКА»**

**Уровень высшего образования  
БАКАЛАВРИАТ**

**Принята на заседании кафедры  
информатики и методики  
преподавания информатики  
(протокол № 11 от «16» июня 2022 г.)**

**Благовещенск 2022**

**СОДЕРЖАНИЕ**

<b>1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....</b>	<b>3</b>
<b>2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ .....</b>	<b>4</b>
<b>3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ) .....</b>	<b>6</b>
<b>4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ .....</b>	<b>7</b>
<b>5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ .....</b>	<b>10</b>
<b>6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА.....</b>	<b>10</b>
<b>7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ .....</b>	<b>19</b>
<b>В ПРОЦЕССЕ ОБУЧЕНИЯ .....</b>	<b>19</b>
<b>8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ .....</b>	<b>20</b>
<b>9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ .....</b>	<b>20</b>
<b>10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА .....</b>	<b>21</b>
<b>11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ .....</b>	<b>23</b>

## 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**1.1 Цель дисциплины** – формирование системы знаний, умений и навыков, связанных с технологиями языками программирования высокого уровня как базы для развития универсальных и профессиональных компетенций, подготовка к использованию как языка программирования, так и методов программирования в учебной и последующей профессиональной деятельности в различных предметных областях.

Основными задачами дисциплины являются

- понимание
  - особенностей различных языков, парадигм и сред программирования;
  - основных принципов процедурного, объектно-ориентированного и функционального программирования;
  - основных принципов использования наиболее распространенных сред программирования;
  - основ процесса процедурного и объектно-ориентированного программирования при решении задач;
- овладение навыками
  - использования основных методов, способов и средств программирования;
  - программистской деятельности в ходе решения прикладных задач;
- развитие умений
  - использования основных типов и структур данных при решении прикладных задач;
  - использования основных методов, способов и средств программирования;

**1.2 Место дисциплины в структуре ООП:** Дисциплина «Программирование» относится к дисциплинам обязательной части блока Б1 (Б1 О.08.01) в предметно-методическом модуле по профилю «Информатика».

Дисциплина «Программирование» использует знания, умения и навыки, сформированные школьным курсом информатики в соответствии со стандартом ФГОС СОО. Освоение программирования необходимо как предшествующее для дисциплин «Практикум для решения предметных задач», «Современные технологии программирования», «Численные методы», «Компьютерное моделирование», «Образовательная робототехника». Дисциплина «Программирование» в профессиональной подготовке выпускника обеспечивает приобретение знаний и навыков, необходимых для применения в будущей профессиональной деятельности при организации процесса обучения базовому и профильному курсу информатики, при поддержке проектной деятельности обучающихся, подготовке их к олимпиадам и конкурсам.

**1.3 Дисциплина направлена на формирование следующих компетенций:** ОПК-9, ПК-2:

– **ОПК-9.** Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности; **индикаторами достижения** которой являются:

- ОПК-9.1 – **имеет** необходимые знания в широком спектре современных информационных технологий.

– **ПК-2.** Способен осуществлять педагогическую деятельность по профильным предметам (дисциплинам модуля) в рамках программ основного общего и среднего общего образования; **индикаторами достижения** которой являются:

- ПК-2.3 – **применяет** методологии программирования и современные информационно-коммуникационные технологии для решения практических задач получения, хранения, обработки и передачи информации;

- ПК-2.6 – **владеет** навыками алгоритмического мышления и приемами написания программ на языках программирования высокого уровня.

**1.4 Перечень планируемых результатов обучения.** В результате изучения дисциплины студент приобретает:

- знание основ современных языков программирования;
- умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умение искать и обрабатывать ошибки в коде;
- умение разбивать решение задачи на подзадачи;
- способность писать грамотный, красивый код;
- способность анализировать как свой, так и чужой код;
- способность работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- способность грамотно строить коммуникацию, исходя из целей и ситуации.

**1.5 Общая трудоемкость дисциплины «Программирование»** составляет 7 зачетных единиц (далее – ЗЕ) (252 часа):

Программа предусматривает изучение материала на лекциях и лабораторных занятиях. Предусмотрена самостоятельная работа студентов по темам и разделам. Проверка знаний осуществляется фронтально, индивидуально, в группе.

#### 1.6 Объем дисциплины и виды учебной деятельности

##### Объем дисциплины и виды учебной деятельности (очная форма обучения)

Вид учебной работы	Всего часов	Семестр 2	Семестр 3
Общая трудоемкость	252	108	144
Аудиторные занятия	108	54	54
Лекции	44	22	22
Лабораторные занятия	64	32	32
Самостоятельная работа	108	54	54
Вид итогового контроля	36	зачет с оценкой	экзамен

## 2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

### 2.1 Очная форма обучения

#### Учебно-тематический план

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные занятия	
<b>Семестр 2</b>					
1.	Введение. Технологии программирования. Инструментальные средства программирования.	8	2	2	4
2.	Переменные, простые типы, операции. Присваивание, ввод, вывод.	12	4	2	6

3.	Базовые алгоритмические конструкции. Ветвления и циклы.	24	4	8	12
4.	Строки.	12	2	4	6
5.	Коллекции: списки, кортежи, словари, множества.	28	4	10	14
6.	Подпрограммы. Функции. Рекурсивные функции.	16	4	4	8
7.	Модули и пакеты.	8	2	2	4
<b>ИТОГО в семестре 2</b>		<b>108</b>	<b>22</b>	<b>32</b>	<b>54</b>
<b>Семестр 3</b>					
8.	Файлы. Работа с файлами.	8	2	2	4
9.	Моделирование различных структур данных. Очередь, дек, стек.	12	2	4	6
10.	Исключения. Обработка исключений	8	2	2	4
11.	Объектно-ориентированное программирование.	48	10	14	24
12.	Программирование графического интерфейса пользователя.	32	6	10	16
	Экзамен	36			
<b>ИТОГО в семестре 3</b>		<b>144</b>	<b>22</b>	<b>32</b>	<b>54</b>
<b>ИТОГО</b>		<b>252</b>	<b>44</b>	<b>64</b>	<b>108</b>

### Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного занятия	Кол-во часов
1.	Введение. Технологии программирования. Инструментальные средства программирования.	Лекция	Лекция-беседа	2
2.	Коллекции: списки, кортежи, словари, множества.	Практикум	Работа в малых группах	6
3.	Исключения. Обработка исключений	Практикум	Работа в малых группах	2
4.	Объектно-ориентированное программирование.	Практикум	Работа в малых группах	6
5.	Программирование графического интерфейса пользователя.	Практикум	Работа в малых группах	4
6.	Программирование графического интерфейса пользователя.	Лекция	Лекция-беседа	2
<b>ИТОГО</b>				<b>22</b>

### 3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)

#### **Тема 1. Введение. Технологии программирования. Инструментальные средства программирования**

История развития языков и технологий программирования. Классификация современных языков программирования высокого уровня. Место языка Python в иерархии языков программирования. Парадигмы программирования: процедурная, функциональная, логическая, объектно-ориентированная. Обзор современных технологий программирования. Инструменты разработки программ.

#### **Тема 2. Переменные, простые типы, операции. Присваивание, ввод, вывод**

Функция print. Переменные. Именованное присваивание. Простые типы данных: int, float, str, bool. Операции над простыми данными. Приоритет выполнения операций. Присваивание значения переменной. Оператор присваивания. Проверка типа данных. Функция input. Преобразование типов данных.

#### **Тема 3. Базовые алгоритмические конструкции. Ветвления и циклы**

Операторы сравнения. Оператор ветвления if-else. Вложенные ветвления, многозначное ветвление. Цикл while. Цикл for. Функции range и enumerate. Операторы continue, break.

#### **Тема 4. Строки**

Тип str. Создание строки. Специальные символы. Кодировки. Операции над строками. Срезы. Функции и методы работы со строками. Функции для работы с символами. Базовые алгоритмы обработки строк.

#### **Тема 5. Коллекции: списки, кортежи, словари, множества**

Списки. Создание списка. Операции над списками. Срезы. Функции и методы работы со списками. Базовые задачи обработки списков. Генераторы списков и списочные выражения. Кортежи. Множества. Создание множества. Операции над множествами. Функции и методы работы с множествами. Словари. Создание словаря. Операции над словарем. Функции и методы работы со словарем. Генераторы словарей.

#### **Тема 6. Подпрограммы. Функции. Рекурсивные функции**

Подпрограмма. Метод нисходящего проектирования программы. Определение функции. Передача аргументов и возвращаемое значение. Параметры функции, позиционные и непозиционные. Вызов функции. Глобальные и локальные переменные, области видимости переменных. Вложенные функции. Рекурсивные функции.

#### **Тема 7. Модули и пакеты**

Хранение функций в модулях. Модули. Инструкция import. Инструкция from. Загрузка модулей. Пакеты. Установка пакетов.

#### **Тема 8. Файлы. Работа с файлами**

Файлы. Типы файлов. Методы для работы с файлами. Доступ к файлам с помощью модуля os.

#### **Тема 9. Моделирование различных структур данных. Очередь, дек, стек**

Структуры данных очередь, стек, дек. Варианты моделирования структур данных.

### **Тема 10. Исключения. Обработка исключений**

Исключения. Классы встроённых исключений. Пользовательские исключения. Инструкция try-except-else-finally. Инструкция with-as.

### **Тема 11. Объектно-ориентированное программирование**

Классы и экземпляры. Метод \_\_init\_\_. Инкапсуляция. Вызов метода. Наследование. Иерархия наследования. Множественное наследование. Специальные методы. Перегрузка операторов. Проектирование и использование классов.

### **Тема 12. Программирование графического интерфейса пользователя**

Графический интерфейс, основы. Библиотека PyQt. Управление окном приложения. Основные компоненты. Обработка сигналов и событий. События клавиатуры и мыши. Диалоговые окна.

## **4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ**

### **4.1 Общие методические рекомендации**

Излагаемая дисциплина предлагает к изучению теоретические программирования и одновременно имеет четкую прикладную направленность. При разработке рабочей программы дисциплины предусмотрено, что определенные вопросы изучаются студентами самостоятельно.

Практикум по дисциплине представлен учебно-методическим материалом по подготовке к занятиям. Дидактические материалы для контроля (самоконтроля) усвоения учебного материала содержат вопросы для подготовки к занятиям и примерные вопросы зачёта и экзамена. Раздел программы «Список литературы и информационных ресурсов» позволяет использовать материалы не только для подготовки к аудиторным занятиям, но и для организации самостоятельной работы, а также для расширения собственных представлений по отдельным разделам изучаемой дисциплины.

Основное предназначение дидактических материалов – помочь студентам организовать самостоятельную подготовку по дисциплине, провести самоконтроль умений и знаний, получить чёткое представление о предстоящих формах контроля.

### **4.2 Методические рекомендации по подготовке к лекциям**

Курс лекций строится на основе четких понятий и формулировок, так, как только при таком подходе студенты приобретают культуру абстрактного мышления, необходимую для высококвалифицированного бакалавра в любой отрасли знаний. Изложение материала должно быть по возможности простым и базироваться на уровне разумной строгости. Изложение теоретического материала дисциплины должно предшествовать выполнению практикума.

Во время лекций по дисциплине студент должен уметь сконцентрировать внимание на рассматриваемых проблемах и включить в работу все виды памяти: словесную, образную и моторно-двигательную. В этом помогает конспектирование сути материала, излагаемого преподавателем (Во время конспектирования в работу включаются зрительная, аудиальная и моторно-двигательная память, позволяющие эффективно усвоить лекционный материал.) Главное, что конспектирование лекции – это не диктант. Для успешной работы студент только выделяет суть, и фиксирует её «своими словами» в объёме, достаточном для гарантированного воспроизведения. Это гораздо более эффективно, чем запись «под диктовку». В ходе возникновения трудностей следует относиться к этому как к признаку правильного хода работы, чётко сформулировать непонимаемый фрагмент вы-

сказывания лектора и задать вопрос, стараясь не нарушать ритм и ход лекции. Часто это помогает всем студентам лучше осознать материал.

Следует быть готовым к тому, что на лекциях периодически проводится опрос студентов по материалам лекций. Подборка вопросов для опроса осуществляется на основе изученного теоретического материала. Такой подход позволяет не только контролировать уровень усвоения теоретического материала, но и организовать эффективный контроль посещаемости занятий на потоковых лекциях и дисциплины в ходе совместной очной работы.

### **4.3 Методические рекомендации по подготовке к лабораторным занятиям**

Наряду с работой на лекциях, ключевое место в учебном процессе занимают лабораторные занятия для апробации, закрепления и переосмысления полученных студентами знаний, содержащих большую долю практического и прикладного характера.

Перед практическим занятием студенту необходимо освежить в памяти теоретический материал по теме практического занятия. Для этого следует обратиться к соответствующим главам учебника, конспекту лекций. Каждое занятие начинается с повторения необходимых элементов теоретического материала по соответствующей теме. Для самопроверки, студенты должны уметь чётко ответить на вопросы, поставленные преподавателем. По характеру ответов преподаватель делает вывод о том, насколько тот или иной студент готов к выполнению упражнений. После такой проверки студентам предлагается выполнить соответствующие задания и варианты задачи.

Порядок решения задач студентами может быть различным. Преподаватель может установить такой порядок, согласно которому каждый студент в отдельности самостоятельно решает задачу без обращения к каким – либо материалам или к преподавателю. Может быть использован и такой порядок решения задачи, когда предусматривается самостоятельное решение каждым студентом поставленной задачи с использованием конспектов, учебников и других методических и справочных материалов. При этом преподаватель обходит студентов, наблюдая за ходом решения и давая индивидуальные указания. По истечении времени, необходимого для решения задачи, один из студентов может быть вызван для её выполнения на доске.

В конце занятия преподаватель подводит его итоги, даёт оценку активности студентов и уровня их знаний, вносит баллы в рейтинговую таблицу.

Каждому студенту необходимо основательно закреплять полученные знания и вырабатывать навыки самостоятельной научной работы. С этой целью в течение семестра студент должен выполнять домашние работы. Часть лабораторных допускается выполнять дома, особенно при опережении графика сдачи, поскольку в процессе сдачи авторство и глубина понимания материала крайне легко проверяется индивидуальными вопросами, к чему тоже следует быть готовым.

Выполнение практикума по дисциплине, задания которого размещены в Электронной информационно-образовательной среде БГПУ, фиксируется и оценивается в СЭО.

### **4.4 Методические указания к самостоятельной работе студентов**

Для успешного усвоения дисциплины необходима правильная организация самостоятельной работы студентов. Эта работа должна содержать:

- регулярную (еженедельную) проработку теоретического материала по конспектам лекций;
- регулярную (еженедельную) подготовку к занятиям.



**Учебно-методическое обеспечение самостоятельной работы  
студентов по дисциплине**

<b>№</b>	<b>Наименование раздела (темы)</b>	<b>Формы/виды самостоятельной работы</b>	<b>Количество часов</b>
1.	Введение. Технологии программирования. Инструментальные средства программирования.	Установка интерпретатора Python. Установка IDE. Знакомство с документацией. Практика работы	8
2.	Переменные, простые типы, операции. Присваивание, ввод, вывод.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	6
3.	Базовые алгоритмические конструкции. Ветвления и циклы.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	14
4.	Строки.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	8
5.	Коллекции: списки, кортежи, словари, множества.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	4
6.	Подпрограммы. Функции. Рекурсивные функции.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	8
7.	Модули и пакеты.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к практическому занятию. Решение задач.	6
8.	Файлы. Работа с файлами.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	4
9.	Моделирование различных структур данных. Очередь, дек, стек.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	6
10.	Исключения. Обработка ис-	Проработка теоретического матери-	4

	ключений	ала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	
11.	Объектно-ориентированное программирование.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	24
12.	Программирование графического интерфейса пользователя.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторному занятию. Решение задач.	16
Итого			108

## 5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

### 5.1 План проведения лабораторных занятий по дисциплине

#### **Инструментальные средства программирования.**

Занятие № 1. Знакомство со средой программирования. Вычисления.

#### **Переменные, простые типы, операции. Присваивание, ввод, вывод.**

Занятие № 2. Линейные алгоритмы. Целочисленная арифметика.

#### **Базовые алгоритмические конструкции. Ветвления и циклы.**

Занятие № 3. Ветвления.

Занятие № 4. Цикл while.

Занятие № 5. Цикл for

Занятие № 6. Сочетание циклов с ветвлениями. Вложенные ветвления.

#### **Строки.**

Занятие № 7. Базовые задачи обработки строк.

Занятие № 8. Функции и методы обработки строк.

#### **Коллекции: списки, кортежи, словари, множества.**

Занятие № 9. Списки

Занятие № 10. Многомерные списки.

Занятие № 11. Кортежи.

Занятие № 12. Множества.

Занятие № 13. Словари.

#### **Подпрограммы. Функции. Рекурсивные функции.**

Занятие № 14. Функции.

Занятие № 15. Рекурсивные функции.

#### **Модули и пакеты.**

Занятие № 16. Модули и пакеты.

#### **Файлы. Работа с файлами.**

Занятие № 17. Файлы.

### **Моделирование различных структур данных. Очередь, дек, стек.**

Занятие № 18. Стек.

Занятие № 19. Очередь. Дек.

### **Исключения. Обработка исключений.**

Занятие № 20. Исключения.

### **Объектно-ориентированное программирование.**

Занятие № 21. Объявление класса

Занятие № 22. Методы классов.

Занятие № 23. Наследование.

Занятие № 24. Переопределение операций.

Занятие № 25. Иерархия классов.

Занятие № 26. Проектирование классов.

Занятие № 27. Проектирование классов. Развитие задачи.

### **Программирование графического интерфейса пользователя.**

Занятие № 28. Библиотека PyQt.

Занятие № 29. QtDesigner.

Занятие № 30. Проектирование интерфейса.

Занятие № 31. Реализация интерфейса.

Занятие № 32. Реализация интерфейса. Развитие задачи.

**Всего: 64 часа**

## **6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА**

### **6.1 Оценочные средства, показатели и критерии оценивания компетенций**

<b>Индекс компетенции</b>	<b>Оценочное средство</b>	<b>Показатели оценивания</b>	<b>Критерии оценивания сформированности компетенций</b>
ОПК-9, ПК-2	Разноуровневые задачи и задания	Низкий (неудовлетворительно)	Выполнение задания студенту не засчитывается если: <ul style="list-style-type: none"> <li>• Задание выполнено менее, чем на половину.</li> </ul>
		Пороговый (удовлетворительно)	Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но: <ul style="list-style-type: none"> <li>• выполнил задание неполно.</li> </ul>
		Базовый (хорошо)	Задание в основном выполнено, но: <ul style="list-style-type: none"> <li>• допущено 1-2 недочета.</li> </ul>
		Высокий (отлично)	Задание выполнено в максимальном объеме.

ПК-2	Экзаменационный вопрос, задача	Низкий (неудовлетворительно)	<p>Ответ студенту не засчитывается если:</p> <ul style="list-style-type: none"> <li>• Задание выполнено менее, чем на половину;</li> <li>• Студент предоставил одно-сложные ответы.</li> </ul>
		Пороговый (удовлетворительно)	<p>Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но:</p> <ul style="list-style-type: none"> <li>• Излагает материал неполно и допускает неточности в определении понятий;</li> <li>• Не умеет достаточно глубоко и доказательно обосновать свои суждения.</li> </ul>
		Базовый (хорошо)	<p>Задание в основном выполнено. Ответы правильные, но:</p> <ul style="list-style-type: none"> <li>• В ответе допущены мало-значительные ошибки и недостаточно полно раскрыто содержание вопроса;</li> <li>• Не приведена четкая полная аргументация решения;</li> <li>• Допущено 1-2 недочета в последовательности и языковом оформлении излагаемого.</li> </ul>
		Высокий (отлично)	<p>Задание выполнено в максимальном объеме. Ответы полные и правильные.</p> <ul style="list-style-type: none"> <li>• Студент приводит четкую полную аргументацию выбранного решения;</li> <li>• Излагает материал последовательно и правильно с точки зрения норм литературного языка.</li> </ul>

## 6.2 Промежуточная аттестация студентов по дисциплине

Промежуточная аттестация является проверкой всех знаний, навыков и умений студентов, приобретённых в процессе изучения дисциплины. Формой промежуточной аттестации по дисциплине является зачёт и экзамен.

Контроль знаний по данной дисциплине проводится в следующих формах:

- контроль выполнения практических заданий;
- решение задач на зачете;
- экзамен.

Для получения зачета в семестре 2, студент обязан выполнить все практические задания и выполнить решение задач на зачете.

### **Критерии оценивания зачёта**

При выставлении оценки учитывается выполнение практикума по дисциплине, задания которого размещены в Электронной информационно-образовательной среде БГПУ.

Оценка «зачтено» ставится, если:

– задания практикума по дисциплине выполнены на 85 и более процентов и зачетные задания решены, возможно с небольшими недочетами.

Оценка «незачтено» ставится, если:

– задания практикума по дисциплине выполнены менее чем на 85 процентов и зачетные задания не решены.

### **Критерии оценивания ответа на зачете с оценкой:**

Оценка «отлично»:

- алгоритм и код программы обеспечивают полное и точное решение поставленной задачи,
- программа снабжена необходимым количеством комментариев,
- код оформлен в соответствии со стандартами,
- дано полное объяснение алгоритма и сделана оценка его эффективности.

Оценка «хорошо»:

- алгоритм и код программы обеспечивают полное и точное решение поставленной задачи,
- комментарии в программе отсутствуют,
- есть недочеты в оформлении кода,
- не дана оценка эффективности алгоритма.

Оценка «удовлетворительно»:

- в алгоритме или коде программы присутствуют ошибки, приводящие к неполному или неточному решению поставленной задачи,
- даны неполные комментарии,
- не дана оценка эффективности алгоритма.

Оценка «неудовлетворительно»:

- программа неверна или отсутствует.

### **Критерии оценивания устного ответа на экзамене:**

- правильность ответа на вопросы (верное, четкое и достаточно глубокое изложение понятий, фактов);
- полнота и одновременно лаконичность ответа;
- новизна учебной информации, степень использования различных источников;
- умение связывать теорию с практикой, творчески применять знания к неординарным ситуациям;
- логика и аргументированность изложения;
- грамотное комментирование, приведение примеров, аналогий;
- культура речи.

Оценка «отлично»:

- полно раскрыто содержание материала в объеме программы;
- четко и правильно даны определения и раскрыто содержание понятий: верно использованы научные термины;
- ответ самостоятельный, использованы ранее приобретенные знания: речь грамотна и логически последовательна.

Оценка «хорошо»:

- раскрыто основное содержание материала;
- ответ самостоятельный;
- определения понятий неполные, допущены незначительные нарушения последовательности изложения, небольшие неточности при использовании научных терминов или в выводах и обобщениях из наблюдений и опытов.

Оценка «удовлетворительно»:

- усвоено основное содержание учебного материала, но изложено фрагментарно, не всегда последовательно;
- определения понятий недостаточно четкие;
- допущены ошибки и неточности в использовании научной терминологии, определении понятий.

Оценка «неудовлетворительно»:

- основное содержание учебного материала не раскрыто;
- не даны ответы на вспомогательные вопросы;
- допущены грубые ошибки в определении понятий, при использовании терминологии;
- ответ на вопрос не дан.

### **6.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов освоения дисциплины**

#### **Варианты разноуровневых задач**

1. В первый день спортсмен пробежал  $x$  километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу  $y$  определите номер дня, на который пробег спортсмена составит не менее  $y$  километров.

**Входные данные**

Программа получает на вход числа  $x$  и  $y$

**Выходные данные**

Программа должна вывести одно натуральное число.

---

**Sample Input:**

10 20

---

**Sample Output:**

9

2. Для праздничного чаепития необходимо купить  $n$  пирожных. В магазине продается всего два вида пирожных, причем пирожных одного вида осталось  $a$  штук, а пирожных другого вида осталось  $b$  штук. Пирожные одного вида считаются одинаковыми. Сколькими способами можно купить ровно  $n$  пирожных?

**Входные данные**

В строке входных данных записаны числа  $n$ ,  $a$  и  $b$  — количество пирожных, которое нужно купить, количество пирожных каждого из двух видов, которые есть в магазине. Все числа — целые, от 1 до 100.

**Выходные данные**

Программа должна вывести одно целое число — количество различных способов купить  $n$  пирожных.

**Примечание**

Например для чисел 5 3 10

Купить 5 пирожных можно 4 способами: 0 пирожных первого вида и 5 пирожных второго вида, 1 пирожное первого вида и 4 пирожных второго вида, 2 пирожных первого вида и 3

пирожных второго вида, 3 пирожных первого вида и 2 пирожное второго вида. Больше способов нет, так как в магазине есть только 3 пирожных первого вида.

**Sample Input:**

5 3 10

**Sample Output:**

4

3. Дорожка замощена плитками в один ряд, плитки пронумерованы числами от 1 до 1000. На плитках с номерами A, B и C ( $A < B < C$ ) сидят три кузнечика, которые играют в чехарду по следующим правилам:

1. На одной плитке может находиться только один кузнечик.
2. За один ход один из двух крайних кузнечиков (то есть с плитки A или с плитки C) может перепрыгнуть через среднего кузнечика (плитка B) и встать на плитку, которая находится ровно посередине между двумя оставшимися кузнечиками (то есть между B и C или A и B соответственно). Если между двумя оставшимися кузнечиками находится чётное число плиток, то он может выбрать любую из двух центральных плиток. Например, если кузнечики первоначально сидели на плитках номер 1, 5, 10, то первым ходом кузнечик с плитки номер 10 может перепрыгнуть на плитку номер 3 (она находится посередине между 1 и 5), или кузнечик с плитки номер 1 может перепрыгнуть на плитку номер 7 или 8 (эти две плитки находятся посередине между плитками 5 и 10).

Даны три числа: A, B, C. Определите, какое наибольшее число ходов может продолжаться игра.

**Входные данные**

Программа получает на вход три целых числа A, B и C ( $1 \leq A < B < C \leq 1000$ ), записанных в строке.

**Выходные данные**

Выведите одно число — наибольшее количество ходов, которое может продолжаться игра.

**Примечание к примеру**

В примере сначала кузнечик с плитки №6 прыгает на плитку №3. Затем кузнечик с плитки №4 прыгает на плитку №2.

**Sample Input:**

1 4 6

**Sample Output:**

2

4. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целочисленные значения от -20 до 20 – сведения о температуре за каждый день ноября. Опишите алгоритм, который находит и выводит максимальную температуру среди дней, когда были заморозки (температура опускалась ниже нуля). Гарантируется, что хотя бы один день в ноябре была отрицательная температура.

**Sample Input:**

2 5 0 -3 -8 -8 -9 -11 -5 0 5 6 -10 -7 0 0 3 5 0 -1

**Sample Output:**

-1

5. Назовём длиной числа количество цифр в его десятичной записи. Например, длина числа 2019 равна 4, а длина числа 7 равна 1. Дан набор из N целых положительных чисел, каждое из которых меньше  $10^9$ . Необходимо определить, числа какой длины чаще всего

встречаются в данном наборе и сколько в нём чисел этой длины. Если числа разной длины встречаются одинаково часто (и чаще чем числа любой другой длины), нужно выбрать большую длину.

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 килобайта и не увеличивается с ростом  $N$ .

#### **Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $1 \leq N \leq 10000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, меньшее, чем  $10^9$ .

#### **Пример входных данных:**

```
5
15
417
125
32
4801
```

#### **Пример выходных данных для приведённого выше примера входных данных:**

```
3 2
```

В данном наборе чаще всего (по 2 раза) встречаются числа длины 2 и 3. Выбираем большую длину, выводим саму длину (3) и количество чисел этой длины (2)

---

#### **Sample Input:**

```
10
8558987
736366971
9331057
9681641
4
5465979
39
49631834
7736
716
```

---

#### **Sample Output:**

```
7 4
```

6. Дан набор из  $N$  целых положительных чисел. Из этих чисел формируются все возможные пары (парой считаются два элемента, которые находятся на разных местах в наборе, порядок чисел в паре не учитывается), в каждой паре вычисляется сумма элементов. Необходимо определить количество пар, для которых полученная сумма делится на 10.

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает одного килобайта и не увеличивается с ростом  $N$ .

#### **Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $1 \leq N \leq 100000$ ).



В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10000.

Пример входных данных:

5  
4  
5  
6  
4  
15

**Пример выходных данных для приведённого выше примера входных данных:**

3

Из 5 чисел можно составить 10 пар. В данном случае у трёх пар сумма делится на 10:  $4 + 6$ ,  $4 + 6$  (в наборе две четвёрки, поэтому пару  $4 + 6$  можно составить двумя способами),  $5 + 15$ .

**Sample Input:**

10  
7  
8324  
97  
3479  
7  
80  
6741  
742  
3489  
59

**Sample Output:**

3

### Примеры задач на зачете

Задание 1.

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое трёхзначных чисел или сообщает, что таких чисел нет (выводит NO). Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести среднее арифметическое трёхзначных чисел или вывести «NO», если таких чисел нет.

Пример работы программы:

Входные данные	Выходные данные
10 120 125 0	122.5
11 1 0	NO

## Задание 2.

Напишите программу, которая в последовательности целых чисел находит сумму и количество чисел, кратных 17, или сообщает, что таких чисел нет. На вход программы подаются целые числа, количество введенных чисел неизвестно, последовательность заканчивается числом 0 (0 – признак окончания ввода, не является членом последовательности). Количество чисел не превышает 100. Введенные числа по модулю не превышают 300. Программа должна вывести сумму и количество чисел, кратных 17, или вывести NO, если таких чисел в последовательности нет.

Пример работы программы:

Входные данные	Выходные данные
34 30 17 0	51 2
-16 5 0	NO

## Задание 3.

Напишите программу для решения следующей задачи.

Участники парусной регаты стартовали одновременно. На финише фиксировалось время прохождения маршрута каждой яхтой (в часах и минутах). Определите время победителя регаты (в часах и минутах). Известно, что соревнования проходили в течение 12 часов. Программа получает на вход количество яхт, принимавших участие в регате  $N$  ( $1 \leq N \leq 100$ ), Затем для каждой яхты вводится два числа: часы и минуты, затраченные на прохождение маршрута.

Пример работы программы:

Входные данные	Выходные данные
2	2 50
3 25	
2 50	

## Задание 4.

Напишите программу для решения следующей задачи.

Девятиклассники участвовали в викторине по математике. Необходимо было ответить на 20 вопросов. Победителем викторины считается участник, правильно ответивший на наибольшее количество вопросов. На сколько вопросов победитель ответил правильно? Если есть участники викторины, которые не смогли дать правильный ответ ни на один из вопросов, выведите YES, иначе выведите NO. Гарантируется, что есть участники, правильно ответившие хотя бы на один из вопросов. Программа получает на вход число участников викторины  $N$  ( $1 \leq N \leq 50$ ), затем для каждого участника вводится количество вопросов, на которые получен правильный ответ.

Пример работы программы:

Входные данные	Выходные данные
4	17

15 12 0 17	YES
---------------------	-----

### Примерный перечень вопросов к экзамену

1. Процедурная парадигма программирования.
2. Базовые типы данных. Переменные, константы и литералы. Приведение типов.
3. Оператор присваивания. Арифметические операторы. Операторы сравнения и логически  
Использование библиотечных функций. Порядок действий (приоритет операторов).
4. Ветвления в программе. Условный оператор и оператор множественного выбора.
5. Операторы цикла.
6. Динамическое распределение памяти в Python.
7. Списки. Базовые задачи обработки списков.
8. Алгоритмы сортировки одномерных массивов.
9. Строковые величины.
10. Статическая, автоматическая и динамическая память.
11. Функции. Описание функции.
12. Формальные и фактические параметры. Передача параметров в функцию по значению, по ссылке, по указателю. Передача массива в функцию.
13. Рекурсия.
14. Классы. Синтаксис объявления классов.
15. Объектно-ориентированная парадигма программирования.
16. Объект. Состояние, поведение, уникальность объекта.
17. Класс. Поля и методы класса.
18. Конструкторы класса.
19. Инкапсуляция.
20. Иерархия классов. Наследование классов.
21. Полиморфизм. Перегрузка и переопределение методов.
22. Виртуальные методы и абстрактные классы.
23. Статические методы.
24. Абстрактные классы и абстрактные методы.
25. Исключения. Обработка исключений.
26. Интерфейсы.
27. Классы коллекции.
28. Объектно-ориентированный подход к проектированию программного обеспечения.
29. Программное обеспечение с графическим пользовательским интерфейсом.
30. Элементы управления.
31. Менеджеры компоновки.
32. События. Обработка событий.

### 7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ В ПРОЦЕССЕ ОБУЧЕНИЯ

**Информационные технологии** – обучение в электронной образовательной среде с целью расширения доступа к образовательным ресурсам, увеличения контактного взаи-

модействия с преподавателем, построения индивидуальных траекторий подготовки, объективного контроля и мониторинга знаний студентов.

В образовательном процессе по дисциплине используются следующие информационные технологии, являющиеся компонентами Электронной информационно-образовательной среды БГПУ:

- официальный сайт БГПУ;
- корпоративная сеть БГПУ;
- система электронного обучения ФГБОУ ВО «БГПУ»;
- электронные библиотечные системы;
- мультимедийное сопровождение лекций и практических занятий;
- цифровые онлайн-инструменты поддержки командной работы.

## **8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

При обучении лиц с ограниченными возможностями здоровья применяются адаптивные образовательные технологии в соответствии с условиями, изложенными в раздел «Особенности организации образовательного процесса по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья» основной образовательной программы (использование специальных учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь и т.п.) с учётом индивидуальных особенностей обучающихся.

## **9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ**

### 9.1 Литература

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2023. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511891> (дата обращения: 28.01.2023).
2. Малов, А. В. Концепции современного программирования : учебное пособие для вузов / А. В. Малов, С. В. Родионов. — Москва : Издательство Юрайт, 2023. — 96 с. — (Высшее образование). — ISBN 978-5-534-14911-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/520338> (дата обращения: 28.01.2023).
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 214 с. — Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/515076> (дата обращения: 28.01.2023).
4. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2023. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519949> (дата обращения: 28.01.2023).
5. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2023. — 219 с. — (Высшее образование). — ISBN 978-5-9916-9983-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511703> (дата обращения: 28.01.2023).

### 9.2 Базы данных и информационно-справочные системы

1. Информационная система «Единое окно доступа к образовательным ресурсам». - Режим доступа: <http://www.window.edu.ru/>
2. Портал научной электронной библиотеки. - Режим доступа: <http://elibrary.ru/defaultx.asp>
3. Сайт Российской академии наук. - Режим доступа: <http://www.ras.ru/>
4. Сайт Института научной информации по общественным наукам РАН. - Режим доступа: <http://www.inion.ru>
5. Сайт Министерства науки и высшего образования РФ. - Режим доступа: <https://minobrnauki.gov.ru>
6. Сайт Министерства просвещения РФ. - Режим доступа: <https://edu.gov.ru/>
7. ХРОНОС - всемирная история в интернете (Исторические источники, Биографический указатель, Генеалогические таблицы, Страны и государства, Религии мира, Исторические организации. Имеются в т.ч. материалы по истории России). - Режим доступа: <http://www.hrono.ru>
8. Русский Биографический Словарь - статьи из Энциклопедического Словаря издательства Брокгауз-Ефрон и Нового Энциклопедического Словаря (включает статьи биографии российских деятелей, а также материалы тома «Россия»). - Режим доступа: <http://www.rulex.ru>
9. People'sHistory - биографии известных людей (история, наука, культура, литература и т.д.). - Режим доступа: <https://www.peoples.ru>

### 9.3 Электронно-библиотечные ресурсы

1. ЭБС «Юрайт». - Режим доступа: <https://urait.ru>
2. Полпред (обзор СМИ). - Режим доступа: <https://polpred.com/news>

## 10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Для проведения лекционных и практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются аудитории, оснащённые учебной мебелью, аудиторной доской, компьютером с установленным лицензионным специализированным программным обеспечением, с выходом в электронно-библиотечную систему и электронную информационно-образовательную среду БГПУ, мультимедийными проекторами, экспозиционными экранами, учебно-наглядными пособиями (мультимедийные презентации).

Для проведения практических занятий также используются компьютерные классы, укомплектованные следующим оборудованием:

- Комплект столов письменных.
- Стол преподавателя.
- Аудиторная доска.
- Компьютеры с установленным лицензионным специализированным программным обеспечением.
- Мультимедийный проектор.
- Экспозиционный экран.
- Учебно-наглядные пособия – мультимедийные презентации по дисциплине «Педагогическая поддержка командной работы школьников».

Самостоятельная работа студентов организуется в аудиториях оснащенных компьютерной техникой с выходом в электронную информационно-образовательную среду вуза, а также в залах доступа в локальную сеть БГПУ.

Лицензионное программное обеспечение: операционные системы семейства Windows, Linux; офисные программы Microsoft office, Libreoffice, OpenOffice; Adobe Photoshop, Matlab, DrWeb antivirus и т.д.

Разработчик: Федченко Г.М. – доцент кафедры информатики и методики преподавания информатики

**11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ**

РПД обсуждена и одобрена для реализации в 2022/2023 уч. г. на заседании кафедры информатики и методики преподавания информатики (протокол № 1 от 21 сентября 2022 г.).

РПД обсуждена и одобрена для реализации в 2024/2025 уч. г. на заседании кафедры информатики и методики преподавания информатики (протокол № 8 от 30 мая 2024 г.).