

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Практикум по прикладной IT-разработке
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Системное программирование и прикладная математика Физтех-школа Прикладной Математики и Информатики центр практик и стажировок ФПМИ
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании центра практик и стажировок ФПМИ 02.04.2024

## Аннотация

В курсе изучаются основы многопоточного программирования и задачи, связанные с ним. Рассматриваются классические сценарии и механизмы синхронизации в многопоточных системах, такие как, например, мьютекс, условная переменная и барьер. Обсуждаются задачи и механизмы разных уровней, которые возникают в языках программирования, поддерживающих возможность асинхронного программирования,

### 1. Цели и задачи

#### Цель дисциплины

- Сформировать у студента понимание основных понятий программного обеспечения.
- Дать студенту теоретические и практические знания о современных технологиях разработки программного обеспечения.
- Дать студенту полную картину разработки проектов с использованием реальных данных и задач.
- Подготовить студента к профессиональной деятельности в области IT-разработок.

#### Задачи дисциплины

- Научить командной работе и управлению проектами.
- Научить разработке проектов с использованием современных технологий, анализа, проектирования и тестирования данных.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные модели памяти, используемые в языках программирования и различных архитектурах процессоров.
- основные алгоритмы и структуры данных.
- принципы разработки программного обеспечения.
- современные технологии и инструменты разработки.

уметь:

- описывать различия моделей памяти.
- программировать на выбранных языках программирования.
- анализировать и решать задачи с использованием программирования.
- проводить тестирование и отладку программного обеспечения.

владеть:

- навыками командной работы и управления проектами.
- навыками документирования кода и проектной документации.
- навыками анализа требований и проектирования программных решений.
- навыками самостоятельного и коллективного развития в области IT разработки.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Введение в практикум по прикладной IT разработке.	4	4		15
2	Основы программирования.	4	4		15
3	Работа с функциями и процедурами в выбранном языке программирования.	4	4		5
4	Основы алгоритмов и структур данных.	4	4		5
5	Работа с файлами и базами данных.	2	2		5
6	Разработка пользовательского интерфейса.	2	2		5
7	Работа с сетевыми технологиями.	2	2		5
8	Тестирование и отладка программного обеспечения.	2	2		5
9	Проектирование и разработка полноценного программного продукта.	2	2		5
10	Защита информации и безопасность в IT разработке.	2	2		5
11	Командная работа и управление проектами.	2	2		5
Итого часов		30	30		75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

##### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 4 (Весенний)

1. Введение в практикум по прикладной IT разработке.

Обзор целей, задач и ожидаемых результатов.

2. Основы программирования.

Переменные, типы данных, операторы, условия, циклы.

3. Работа с функциями и процедурами в выбранном языке программирования.

Работа с функциями и процедурами является одним из основных аспектов прикладной IT разработки. В различных языках программирования существуют различия в синтаксисе и подходах к созданию и использованию функций и процедур, но основные принципы остаются общими.

Например, в языке Python функции объявляются с помощью ключевого слова `def`, аргументы передаются в функцию в скобках, а результат возвращается с помощью ключевого слова `return`. Пример:

```
Python
def add_numbers(a, b):
    return a + b

result = add_numbers(3, 5)
print(result) # Выведет: 8
```

В языке Java функции объявляются как методы внутри классов. Пример:

```
Java
public class MathUtils {
    public static int addNumbers(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int result = MathUtils.addNumbers(3, 5);
        System.out.println(result); // Выведет: 8
    }
}

def add_numbers(a, b):
    return a + b

result = add_numbers(3, 5)
print(result) # Выведет: 8
```

#### 4. Основы алгоритмов и структур данных.

Массивы, списки, стеки, очереди.

#### 5. Работа с файлами и базами данных.

Чтение, запись, обновление данных.

#### 6. Разработка пользовательского интерфейса.

Графический дизайн, взаимодействие с пользователем.

#### 7. Работа с сетевыми технологиями.

Создание клиент-серверных приложений, обмен данными по сети.

#### 8. Тестирование и отладка программного обеспечения.

Методы тестирования, поиск и исправление ошибок.

9. Проектирование и разработка полноценного программного продукта.

Анализ требований, проектирование архитектуры, реализация функционала.

10. Защита информации и безопасность в IT разработке.

Основные принципы безопасности, защита данных и приложений.

11. Командная работа и управление проектами.

Распределение задач, контроль версий, коммуникация в команде.

## **5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Учебная аудитория, оснащенная компьютерами для каждого студента.

## **6. Перечень рекомендуемой литературы**

Основная литература

1. Программирование на C++ [Электронный ресурс] / Н. Дейл, Ч. Уимз, М. Хедингтон. — М., ДМК Пресс, 2007.— URL: <https://e.lanbook.com/book/1219> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература

## **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

- "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
- "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
- "The Pragmatic Programmer: Your Journey to Mastery" by Andrew Hunt, David Thomas

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

учебная аудитория, экран и проектор.

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

На занятиях особое внимание обращается на современные способы разработки программных продуктов. Необходимо требовать от студентов качественного и понятного программного кода на ряду и в равной степени с правильной работой программы. На занятиях стоит постоянно разбирать ошибки, которые допускали студенты в своих программах, а также логические ошибки.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Системное программирование и прикладная математика Физтех-школа Прикладной Математики и Информатики центр практик и стажировок ФПМИ
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

**Разработчик:** О.Н. Ивченко, заведующий кафедрой

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Практикум по прикладной IT-разработке» обучающийся должен:

### знать:

- основные модели памяти, используемые в языках программирования и различных архитектурах процессоров.
- основные алгоритмы и структуры данных.
- принципы разработки программного обеспечения.
- современные технологии и инструменты разработки.

### уметь:

- описывать различия моделей памяти.
- программировать на выбранных языках программирования.
- анализировать и решать задачи с использованием программирования.
- проводить тестирование и отладку программного обеспечения.

### владеть:

- навыками командной работы и управления проектами.
- навыками документирования кода и проектной документации.
- навыками анализа требований и проектирования программных решений.
- навыками самостоятельного и коллективного развития в области IT разработки.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Создание простого веб-приложения.
2. Разработка мобильного приложения.
3. Работа с большими данными.
4. Разработка игрового приложения.
5. Что такое функция и процедура в контексте программирования?
6. Как объявляются функции в языке программирования Python? Как передаются аргументы и возвращается результат?
7. Как объявляются методы (функции) в Java? Какие модификаторы доступа могут использоваться для методов?
8. Какие преимущества предоставляет использование функций и процедур при разработке программного обеспечения?
9. Чем отличается функция от процедуры?
10. Какие типы данных могут быть переданы в функцию или процедуру? Какие типы данных могут быть возвращены из функции?

#### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

##### 1. Общие вопросы:

- Что такое жизненный цикл разработки программного обеспечения и какие этапы он включает?
- Какие принципы объектно-ориентированного программирования вы знаете и как они применяются в разработке ПО?
- В чем разница между front-end и back-end разработкой? Какие технологии используются в каждом из этих направлений?

##### 2. Вопросы по веб-разработке:

- Что такое HTML, CSS и JavaScript? Какие функции выполняют эти языки в веб-разработке?
- Какие принципы адаптивного и отзывчивого дизайна вы знаете? Почему это важно для веб-разработки?
- Какие инструменты и технологии используются для работы с базами данных в веб-разработке?

##### 3. Вопросы по мобильной разработке:

- Какие основные платформы для мобильной разработки существуют? Какие языки программирования используются для создания мобильных приложений?
- Что такое API и как оно используется в мобильной разработке? Приведите примеры API, которые могут быть полезны для мобильных приложений.
- Какие основные принципы безопасности следует учитывать при разработке мобильных приложений?

##### 4. Вопросы по анализу данных и машинному обучению:

- Что такое машинное обучение и какие типы задач можно решать с его помощью?
- Какие библиотеки и инструменты используются для анализа данных и построения моделей машинного обучения?
- Какие методы визуализации данных вы знаете и как они помогают в анализе данных?

#### Критерии оценивания

- оценка «отлично (10)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (6)» выставляется студенту, если он знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (5)» выставляется студенту, если он знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;



- оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.