

# ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ ПРОГРАММИРОВАНИЯ

## 1. Цель освоения дисциплины

Сформировать у будущего учителя информатики систему специальных компетенций в области разработки программных средств для решения практических задач получения, хранения, обработки и передачи информации в профессиональной деятельности.

## 2. Место дисциплины в структуре ОПОП

Дисциплина «Высокоуровневые методы программирования» относится к вариативной части блока дисциплин и является дисциплиной по выбору.

Для освоения дисциплины «Высокоуровневые методы программирования» обучающиеся используют знания, умения, способы деятельности и установки, сформированные в ходе изучения дисциплин «Информационные технологии», «Компьютерная графика», «Офисные технологии», «Программирование», «Теория чисел и числовые системы».

Освоение данной дисциплины является необходимой основой для последующего изучения дисциплин «Актуальные проблемы информатики и образования», «Архитектура компьютера», «Информационные системы», «Информационные технологии в управлении образованием», «Компьютерное моделирование», «Методы и средства защиты информации», «Операционная система Linux», «Основы искусственного интеллекта», «Основы робототехники», «Перспективные направления искусственного интеллекта», «Перспективные направления компьютерного моделирования», «Построение Windows-сетей», «Практикум по решению задач на ЭВМ», «Программные средства информационных систем», «Проектирование информационных систем», «Разработка Flash-приложений», «Разработка интернет-приложений», «Современные языки программирования», «Специализированные математические пакеты», «Теоретические основы информатики», «Эксплуатация компьютерных систем», прохождения практики «Преддипломная практика».

## 3. Планируемые результаты обучения

В результате освоения дисциплины выпускник должен обладать следующими компетенциями:

– готовностью применять предметные и метапредметные знания фундаментальной и прикладной информатики для решения теоретических и практических задач, реализации аналитических и технологических решений в области представления и обработки информации, информатизации образования (СК-1).

**В результате изучения дисциплины обучающийся должен:**

### **знать**

- основы современных методологий программирования;
- основы современных технологий разработки программного обеспечения;
- принципы объектно-ориентированного программирования;

### **уметь**

- применять методы декомпозиции и абстракции при разработке программ;
- создавать программы в современных средах RAD;
- применять полученные знания при решении практических задач профессиональной деятельности;

### **владеть**

- навыками отладки и тестирования программ;
- навыками объектно-ориентированного программирования.

#### **4. Общая трудоёмкость дисциплины и её распределение**

количество зачётных единиц – 3,

общая трудоёмкость дисциплины в часах – 108 ч. (в т. ч. аудиторных часов – 16 ч., СРС – 88 ч.),

распределение по семестрам – 2 курс, зима,

форма и место отчётности – зачёт (2 курс, зима).

#### **5. Краткое содержание дисциплины**

Понятие методологии программирования.

Атрибуты методологии программирования. Парадигма программирования. Базовые программные единицы в парадигме программирования. Декомпозиция и абстракция в программировании. Принципы и правила декомпозиции. Виды абстракции. Основные ядра методологий программирования.

Новейшие направления и стандарты в области технологии программирования.

Современные технологии разработки программного обеспечения. Законы эволюции программного обеспечения. Жизненный цикл программных средств в соответствии с подходом RAD. Программирование в средах современных информационных систем. Особенности программирования в оконных операционных средах. Основные стандартные модули, обеспечивающие работу в оконной операционной среде. Среда разработки: система окон разработки; система меню. Отладка и тестирование программ.

Объектно-ориентированное проектирование и программирование.

Объектно-ориентированное проектирование. Язык C# и платформа .NET. Детали проектирования и построения класса. Поля, методы, свойства. Модификаторы доступа.

Конструкторы. Отношения между классами. Клиенты и наследники. Наследование и встраивание. Иерархия классов. Полиморфизм. Sealed – классы. Интерфейсы.

Множественное наследование. Абстрактные классы. Интерфейсы. Сигнатура. Делегаты и анонимные методы. Классы с событиями.

#### **6. Разработчик**

Глазов Сергей Юрьевич, кандидат физико-математических наук, доцент кафедры физики, методики преподавания физики и математики, ИКТ ФГБОУ ВО «ВГСПУ».