

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, КУЛЬТУРЫ
И ИССЛЕДОВАНИЙ РЕСПУБЛИКИ МОЛДОВА

НАЦИОНАЛЬНЫЙ КУРРИКУЛУМ

КУРРИКУЛУМНАЯ ОБЛАСТЬ
«ТЕХНОЛОГИИ»

ДИСЦИПЛИНА ПО ВЫБОРУ

**ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ
НА C / C++**

VII–XII классы

Кишинев, 2020

Утвержден:

- Национальным Консилиумом по Куррикулуму, протокол №. 25 от 25.06.2020
- Приказ Министерства Образования, Культуры и Исследований №. 653/2020

Координатор:

Анжела Присакару, главный консультант, Министерство Образования, Культуры и Исследований

РАБОЧАЯ ГРУППА:

Серджиу Корлат, магистр точных наук, высшая дидактическая степень, Технический Университет Молдовы, координатор;

Григорий Василаке, высшая дидактическая степень, директор лицея “Mircea Eliade”, преподаватель информатики.

Состав рабочей группы одобрен Приказом Министерства Образования, Культуры и Исследований № 431/2020.

Перевод: Серджиу Корлат, 2021

Куррикулум разработан при поддержке проекта „*Tekwill în Fiecare Școală/TwentyTU*”, в рамках Меморандума о сотрудничестве между Министерством Образования, Культуры и Исследований, Министерством Экономики и Инфраструктуры, Национальной Ассоциацией ИКТ Компаний и Общественным Учреждением - Центром обучения и инноваций в области ИКТ - Tekwill по внедрению цифрового образования на национальном уровне, подписанном 14 июня 2018.

Введение:

Куррикулум дисциплины по выбору «Программирование алгоритмов на С / С++» является нормативным и обязательным документом для реализации процесса преподавания - обучения на гимназическом и лицейском образовательных уровнях. Курс «Программирование алгоритмов на С / С++» является дополнительной дисциплиной по выбору в рамках учебной дисциплиной Информатика, для гимназического и лицейского образования.

Предмет изучения: основы программирования, применения и разработки алгоритмов.

Цели:

- Когнитивный контекст: курс разработан с целью способствования раннему формированию у учащихся алгоритмического мышления посредством:
 - Изучения алгоритмов в историческом, научном и прикладном контексте;
 - Анализа ИТ-инструментов для реализации алгоритмов;
 - Анализа и применения алгоритмических моделей для решения задач, традиционно связанных с другими областями знаний или деятельности, кроме информатики;
 - Проектирования алгоритмов.
- Педагогический контекст: курс разработан с целью расширения применения алгоритмических моделей в обучении посредством:
 - расширения спектра знаний, адаптированных к возрасту учеников, преподаваемых в рамках школьных дисциплин;
 - содействия междисциплинарной образовательной деятельности;
 - адаптации методов преподавания – обучения к алгоритмическим формам обучения;
 - равномерного распределения информационной нагрузки и преемственностью между классами и образовательными ступенями;
 - дифференциации и индивидуализации преподавания - обучения;
 - формирования минимального набора предварительных условий для углубленного изучения Информатики

Функции куррикулума школьной дисциплины по выбору «Программирование алгоритмов на С / С++»:

- нормативный акт процесса преподавания-обучения-оценки в контексте педагогики, основанной на компетенциях;
- ориентир для дидактического планирования и реализации образовательного процесса с точки зрения педагогики, основанной на компетенциях;
- базовый компонент для разработки стратегий оценивания в рамках курса;
- ориентирование образовательного процесса к формированию компетенций учащихся;
- фундаментальный компонент для разработки школьных учебников в традиционном и в цифровом формате, методических пособий, оценочных тестов.

Целевые группы: куррикулум курса предназначен для преподавателей, авторов учебников и цифровых образовательных ресурсов, студентов и других категорий заинтересованных лиц.

Дидактическая концепция курса

Определение школьной дисциплины по выбору Программирование алгоритмов на C / C++. Предметом изучения курса является компьютерное моделирование процессов различной природы (технической, социальной, биологической и т. д.) через призму их алгоритмизации и последующего описания с использованием процедурных языков программирования высокого уровня.

Курс разработан в поддержку усилий обязательных школьных предметов в области математики и естественных наук (STEAM) по формированию аналитического мышления учащихся, развитию логики, как общей, так и по специализированным областям знаний. Представляет собой дополнительную возможность усиления специфических компетенций школьной дисциплины Информатика, являясь одновременно и средой для самообучения учащихся, образовательная траектория которых стремится к профессиональной сфере ИКТ.

Образовательная значимость курса состоит в:

- расширении учебной модели изучения базовых концептов информатики, которые включают в себя элементы логики, алгоритмизации, моделирования и программирования природных процессов и человеческой деятельности;
- развитию практических навыков использования компьютерных моделей для оптимизации операций по обработке информации;

Специфические принципы преподавания-обучения в рамках курса. Куррикулум дисциплины по выбору Программирование алгоритмов на C / C++ предлагает модулярную модель изучения данной дисциплины. Эта модель способствует гибкому подходу к изучению курса для учащихся, которые в начале процесса обучения обладают компетенциями, достаточными для старта из точки сочленения модулей курса, с последующим углубленным изучением алгоритмической компоненты и конкурсной информатики. В то же время курс расширяет кругозор учащихся относительно методов реализации концепций информатики для решения проблем, порождаемых современным обществом. В этом контексте выделяются следующие специфические свойства курса:

Модульный подход к курсу – структурирование содержимого в модульную, гибкую модель, которая направлена на развитие навыков учащегося мыслить алгоритмически и использовать инструменты компьютерного моделирования.

Центрирование учебной деятельности на ученике – принятие модели активного обучения, сосредоточенного на учащихся; ориентированной на индивидуальные или групповые мероприятия, которые позволяли бы развитие независимости действий, оригинальности, креативности, способности работы в команде - совмещая вышеперечисленное с индивидуализацией ритма обучения – подобно модели, используемой для школьной дисциплины Информатика.

Обеспечение социальной функциональности учебного процесса – развитие навыков и компетенций, необходимых для органической интеграции учащихся в информационное общество. Принцип реализуется посредством практической деятельности по изучению и использованию алгоритмов и элементов языков программирования высокого уровня.

Обеспечение междисциплинарных связей – подход к реализации сложных моделей, связанных с другими школьными предметами, разработка и внедрение алгоритмов для выявления решений проблем в самых разных областях.

Общие указания по преподаванию-обучению в рамках курса. Общий процесс преподавания – обучения - оценки в рамках курса «Программирование алгоритмов на С / С++» реализуется в контексте системы компетенций для среднего и среднего образования, с дифференциацией содержания, специфичного для каждого уровня обучения.

Ключевые трансверсальные компетенции

- Компетенции обучения / научиться учиться.
- Компетенции общения на румынском языке.
- Компетенции общения на иностранных языках.
- Действенно-стратегические компетенции.
- Компетенции самопознания и самореализации
- Межличностные, гражданские и нравственные компетенции.
- Базовые компетенции по математике, наукам и технологиям.
- Цифровые компетенции в области информационных и коммуникационных технологий (ИКТ).
- Общекультурные и межкультурные компетенции (восприятия и создания ценностей)
- Предпринимательские компетенции.

Транс дисциплинарные компетенции

Компетенции обучения / научиться учиться:

- владение методологией получения и интеграции знаний о природе, человеке и обществе для удовлетворения потребностей, а также их применения для улучшения качества личной и общественной жизни.

Компетенции общения на румынском языке:

- аргументированное общение на румынском языке в конкретных жизненных ситуациях;
- научно обоснованное общение.

Компетенции общения на иностранных языках:

- знание иностранного языка;
- аргументированное общение на иностранном языке в конкретных жизненных ситуациях.

Базовые компетенции по математике, наукам и технологиям:

- организация личной деятельности в условиях постоянно совершенствующихся технологий;
- получение и применение фундаментальных знаний в области математики, естественных наук и технологий в соответствии с личными потребностями;
- предложение новых идей в различных научных областях.

Действенно-стратегические компетенции:

- планирование своей деятельности, предвидение конечного результата, предложение решений в проблемных ситуациях из различных областей;
- самостоятельный и творческий подход в различных жизненных ситуациях для защиты окружающей среды.

Цифровые компетенции в области информационных и коммуникационных технологий (ИКТ).

- Использование, в конкретных жизненных ситуациях, цифровых устройств;
- Создание документов в области связи и информации, а также использование в реальных ситуациях цифровых сервисов, включая Интернет.

Межличностные, гражданские и нравственные компетенции:

- сотрудничество в группе/команде, предотвращение конфликтных ситуаций и уважение мнения других людей;
- проявление активной жизненной позиции, гражданской солидарности и согласия за недискриминационное общество;
- действие в разных жизненных ситуациях на основании нравственных норм и духовных ценностей.

Компетенции самопознания и самореализации:

- критическое осознание своих действий в целях постоянного совершенствования и личностной самореализации;
- ответственное отношение к здоровому образу жизни;
- адаптирование к новым условиям и ситуациям.

Общекультурные и межкультурные компетенции (восприятия и создания ценностей):

- знание ценностных систем национальной культуры и культуры других национальностей с целью их творческого применения и личной самореализации;
- толерантность к межкультурным ценностям.

Предпринимательские компетенции:

- знания и навыки самореализации в сфере предпринимательства в условиях рыночной экономики;
- сознательный выбор области профессиональной деятельности в будущем.

Ключевые компетенции, развиваемые в рамках курса «Программирование алгоритмов на C / C++»

Непосредственно развиваемые

Базовые компетенции по математике, наукам и технологиям – приобретение и использование базовых знаний в области математики, естественных наук и технологий для решения задач и повседневных ситуаций, для продолжения технического или высшего образования по окончании лица.

Действенно-стратегические компетенции – выявление действенно-стратегических проблем и предложение решений. Планирование собственной деятельности и прогнозирование ожидаемых результатов. Разработка собственных планов и стратегий индивидуальной и групповой деятельности.

Цифровые компетенции в области информационных и коммуникационных технологий (ИКТ) – использование в реальных ситуациях цифровых инструментов для общения, сбора, хранения и обработки информации. Создание электронных документов. Использование основных электронных услуг (электронное правительство, электронный бизнес, электронное образование, электронное здравоохранение, электронная культура).

Косвенно развиваемые

Межличностные, гражданские и нравственные компетенции – командная работа.

Компетенции обучения / научиться учиться – планирование и организация собственного обучения индивидуально и в группе.

Компетенции общения на иностранных языках – общение на иностранном языке в новых ситуациях.

Специфические компетенции, развиваемые в рамках курса «Программирование алгоритмов на C / C++»

1. Использование средств ИКТ для обработки данных.
2. Научное восприятие роли и влияния компьютерных технологий в современном обществе, знание принципов и законов работы цифрового оборудования.
3. Разработка компьютерных моделей объектов, систем и процессов, часто встречающихся в повседневной деятельности.
4. Применение средств алгоритмизации, формализации, анализа, синтеза и компьютерной реализации инструментария, необходимого для решения проблем, связанных с цифровой обработкой информации в повседневной деятельности.
5. Проведение виртуальных экспериментов, решение задач повседневной деятельности и разработка моделей изучаемых явлений с помощью цифровых приложений, лабораторий и образовательных сред; интерпретация полученных результатов.
6. Соблюдение нормативно-правовой базы, правил безопасности, эргономических и этических норм при использовании средств ИКТ.

Администрирование учебной дисциплины

Статус	Куррикулярная область	Образовательная ступень	Количество модулей	Количество часов
По выбору	Математика и Науки	Гимназия, лицей	2	34

Распределение часов по модулям

Модуль I: Элементы программирования			
Учебные единицы	Часы		примечания
	Гимназия	Лицей	
Что такое программирование	1	1	
Среды разработки программ	1	1	
Данные и переменные	1	1	
Структура программы. Ввод и вывод	2	1	
Программы с линейной структурой. Арифметические операции и выражения	2	1	
Команды выбора. Логические операции и операции сравнения.	2	1	
Множественный выбор.	1	1	
Повторение действий	1	1	
Циклы со счетчиком	1	1	
Циклы, управляемые условием	1	1	
Линейные массивы	1	1	
Двумерные массивы	1	1	
Строки символов. Операции со строками	2	1	
Использование подпрограмм	-	1	
Функции	-	2	
Глобальные и локальные переменные	-	1	
Всего часов (Модуль 1)	17	17	

Модуль II: Алгоритмы. Введение

Учебные единицы	Часы		примечания
	Гимназия	Лицей	
Алгоритмы. История	1	1	
Способы описания алгоритмов	1	1	
Элементарные алгоритмы	2	1	
Поиск в массивах	1	1	
Упорядочивание предметов. Алгоритм BubbleSort	2	1	
Другой алгоритм сортировки: Сортировка выбором	1	1	
Что означает полный перебор? Задачи, решаемые полным перебором	2	1	
Рекурсия	-	1	
Волновой алгоритм	-	1	
Рекурсивные методы: Разделяй и Властвуй	1	1	
Рекурсивные методы: Перебор с возвратом	1	1	
Графы. Определения. Способы представления	1	1	
Графы. Поиск в графе	1	1	
Остовные деревья минимальной стоимости	1	1	
Кратчайшие расстояния в графе	-	1	
Эвристические алгоритмы. Жадные алгоритмы	2	1	
Эвристические алгоритмы. Игра «Жизнь»	-	1	
Всего (Модуль 2)	17	17	
Всего	34	34	

Примечания:

- Распределение часов по темам является примерным.
- Порядок тем может быть изменен, если не затронута научная или дидактическая логика.

Модуль I: Элементы программирования

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
Учебная единица 1: Элементы программирования		
<ul style="list-style-type: none"> • знание основных понятий, ассоциируемых процессу программирования • идентификация программных продуктов и их элементов • описание этапов эволюции языков программирования • объяснение процессов, связанных с созданием программы 	<p>Программа</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Команды • Алгоритмы • Программы • Языки программирования <ol style="list-style-type: none"> 1. Поколения 2. Язык С 3. Развитие С++ 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Описанию областей использования программ • Аргументации необходимости использования программ • написанию программы «Hello, World!» и подобных • компиляции написанных программ • исправлению ошибок в программах • запуску программ
<ul style="list-style-type: none"> • описание этапов разработки программы на языках программирования С/С • Использование сред разработки для создания программ 	<p>Среды разработки программ</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Этапы разработки программы. <ul style="list-style-type: none"> ○ Написание кода ○ Устранение ошибок ○ Компилирование • Среда разработки <ul style="list-style-type: none"> ○ свойства ○ пример 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Идентификации сред разработки программ для языков С/С++, присутствующих на локальном компьютере • Доступу, регистрации и запуску сред веб-разработки • Определению основных компонентов контроля сред разработки • Настройке интерфейса среды разработки • Написанию кода и устранению неполадок в средах разработки программ
Учебная единица 2: Данные и действия		
<ul style="list-style-type: none"> • Распознавание основных понятий, ассоциированных данным 	<p>Данные и переменные</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Данные. 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Выбору соответствующего типа данных.

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> • Ассоциирование адекватных типов для числовых и символьных значений • Объявление переменных • Определение констант 	<ul style="list-style-type: none"> • Простые типы данных: <ul style="list-style-type: none"> ○ int ○ float ○ char • Идентификаторы • Переменные • Константы 	<ul style="list-style-type: none"> • Объявлению переменных • Определению констант • Непосредственному назначению значений заявленным переменным
<ul style="list-style-type: none"> • Вывод сообщений и констант • Присваивание значений простым переменным в соответствии с их типом 	<p>Вывод и ввод данных Содержание:</p> <p>Ввод / вывод данных в программах C:</p> <ul style="list-style-type: none"> ○ printf ○ scanf ○ спецификаторы формата <ul style="list-style-type: none"> • Ввод / вывод данных в программах C++: <ul style="list-style-type: none"> ○ cin >> ○ cout << 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Присваиванию переменным значений с помощью команд ввода (scanf (C), cin (C++)) • Выводу значений переменных (printf (C), cout (C++)) • Форматированному выводу значений переменных
<ul style="list-style-type: none"> • Перечень операций, применимых к числовым переменным • Описание арифметических выражений • Назначение результатов переменной 	<p>Арифметические операции и выражения</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Операции <ul style="list-style-type: none"> ○ присваивание ○ + - * / % • Выражения <ul style="list-style-type: none"> ○ Приоритет операций ○ Скобки 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Присваиванию переменным значений путем вычисления простых выражений • Выводу вычисленных значений выражений • Форматированному выводу вычисленных значений выражений • Созданию сложных арифметических выражений <p>Задачи по:</p> <ul style="list-style-type: none"> • Выполнению элементарных вычислений • Вычислению площади, объема геометрических фигур, временных интервалов.

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
Учебная единица 3: Операторы выбора		
<ul style="list-style-type: none"> • Описание концепции ветвления в программе • Создание реляционных выражений • Вставка реляционных выражений в операторы выбора • Разработка программ, содержащих операторы выбора 	<p>Операторы выбора. Операции сравнения</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Реляционные операции <ul style="list-style-type: none"> ○ <, >, <=, >=, ==, != • оператор выбора if ○ Логическая схема выполнения оператора ○ Выбор исполняемой ветви оператора 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Созданию и вычислению простых реляционных выражений • Созданию и вычислению комплексных реляционных выражений <p>Задачи по:</p> <ul style="list-style-type: none"> • Определению максимального / минимального значения • Проверке четности / делимости • Моделированию арифметических калькуляторов • Множественным преобразованиям
<ul style="list-style-type: none"> • Идентификация логических операций • Описание комплексных реляционных выражений используя инструментов языка программирования • Использование операторов выбора для проверки составных условий 	<p>Логические операции и составной выбор</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Логические операции <ul style="list-style-type: none"> ○ &&, ○ , ○ !. • Логические выражения • Таблицы истинности • Вложенные команды выбора 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Созданию и вычислению простых логических выражений • Созданию и вычислению комплексных логических выражений • Описанию вложенных условий <p>Задачи по:</p> <ul style="list-style-type: none"> • Определению типа треугольника по значениям величин углов • Моделированию процесса принятия комплексных решений в повседневной жизни
<ul style="list-style-type: none"> • Мотивирование необходимости использования операторов множественного выбора • Описание набора ветвей множественного выбора 	<p>Множественный выбор</p> <p>Содержание:</p> <p>Оператор множественного выбора (switch)</p> <ul style="list-style-type: none"> ○ Логическая схема оператора ○ Селектор <ul style="list-style-type: none"> ▪ переменная ▪ выражение ○ Ветви выбора 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Созданию и вычислению выражений множественного выбора • Идентификации набора значений и действий селектора <p>Задачи по:</p> <ul style="list-style-type: none"> • Множественным преобразованиям

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> Интеграция операторов множественного выбора в программный код 		<ul style="list-style-type: none"> Моделированию процесса принятия комплексных решений в повседневной жизни
Учебная единица 4: Циклы		
<ul style="list-style-type: none"> Описание понятия цикла в программировании выявление ситуаций, связанных с использованием циклов объяснение модели работы циклов со счетчиком / с условием 	<p>Повторение действий</p> <p>Содержание:</p> <ul style="list-style-type: none"> повторение действий <ul style="list-style-type: none"> циклы со счетчиком Логическая схема цикла, управляемого счетчиком Циклы с условием Логическая схема цикла, управляемого условием 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Моделированию циклов со счетчиком в арифметических выражениях Моделированию циклов, управляемых условием в арифметических выражениях Описанию ситуаций повседневной жизни, требующих использования циклов, управляемых условием или счетчиком
<ul style="list-style-type: none"> Объяснение концепции цикла со счетчиком Описание структуры цикла со счетчиком Идентификация алгоритмов / действий, которые могут быть выполнены с помощью циклов со счетчиком Разработка программ с циклами со счетчиком 	<p>Циклы со счетчиком</p> <p>Содержание:</p> <ul style="list-style-type: none"> Циклы со счетчиком Структура оператора for <ul style="list-style-type: none"> Счетчик Начальное значение / выражение Конечное значение / выражение Шаг изменения Вложенные циклы. 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Проверке правильности описания оператора цикла Описанию операторов цикла с помощью инструментария языка программирования Оценке результатов, отображаемых программами, содержащими операторы цикла Преобразованию программ с помощью операторов цикла <p>Задачи по:</p> <ul style="list-style-type: none"> Расчету арифметических выражений: количества, прогрессии, факториала и т. д. Генерации замечательных числовых последовательностей: Фибоначчи, квадратов, простых чисел, пр. «Рисованию» в консоли

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> Объяснение концепции цикла, управляемого условием Описание структуры цикла управляемого пред(пост) условием Определение алгоритмов / действий, которые могут быть выполнены с помощью цикла, управляемого пред(пост) условием Разработка программ с циклами, управляемыми условием 	<p>Циклы, управляемые условием</p> <p>Содержание:</p> <ul style="list-style-type: none"> Циклы, управляемые условием While (предусловие) Do – while (постусловие) Структура оператора while <ul style="list-style-type: none"> Вычисляемое выражение Условие повторения Тело цикла Структура оператора do - while Вложенные циклы. 	<ul style="list-style-type: none"> Выполнению расчетов на шахматной доске и т. д. <p>Упражнения по:</p> <ul style="list-style-type: none"> Проверке правильности условных циклов Описанию условных циклов, используя инструментальный язык программирования Вычислению результатов, отображаемых программами, содержащими условные циклы Преобразованию программ с помощью операторов цикла циклу <p>Задачи по:</p> <ul style="list-style-type: none"> Вычислению суммы цифр в целых числах Условному повторению вычислений. Проверке корректности введенных данных Управляемого рисования в консоли
Учебная единица 5: Структуры данных		
<ul style="list-style-type: none"> Объяснение понятия структуры данных Описание структуры линейного массива Идентификация элементов линейного массива Объявление линейных массивов в качестве переменных Использование линейных массивов для хранения и обработки числовых данных 	<p>Линейные массивы</p> <p>Содержание:</p> <ul style="list-style-type: none"> Линейные массивы <ul style="list-style-type: none"> Объявление Присваивание значений Прохождение Вывод Операции над линейными массивами <ul style="list-style-type: none"> Доступ к значениям элементов по индексу Поиск элементов с данным значением 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Проверке правильности объявления массивов Проверке правильности ввода данных в массивы Вычислению результатов, отображаемых программами, содержащими линейные массивы Преобразованию программ используя линейные массивы <p>Задачи по:</p> <ul style="list-style-type: none"> Расчету результатов арифметических операций на подмножествах элементов массива. Поиску элемента с заданными свойствами в массиве Добавлению/замене элементов в массиве

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
		<ul style="list-style-type: none"> • Перемещению элементов массива • Сортировке значений элементов массива по заданному критерию • Поиску последовательностей значений с заданными свойствами, и так далее.
<ul style="list-style-type: none"> • Описание структуры двумерного массива • Идентификация элементов двумерного массива • Объявление двумерных массивов в качестве переменных • Использование двумерных массивов для анализа числовых данных 	<p>Двумерные массивы Содержание:</p> <ul style="list-style-type: none"> • Двумерные массивы <ul style="list-style-type: none"> ○ Объявление ○ Присваивание значений ○ Вывод • Операции над двумерными массивами <ul style="list-style-type: none"> ○ Доступ к значениям элементов по индексу ○ Поиск элементов с данным значением • Квадратные двумерные массивы <ul style="list-style-type: none"> ○ Главная диагональ ○ Дополнительная диагональ 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Проверке правильности объявления двумерных массивов • Проверке правильности ввода данных в массивы • Вычислению результатов, отображаемых программами, содержащими двумерные массивы • Преобразованию программ используя двумерные массивы <p>Задачи по:</p> <ul style="list-style-type: none"> • Поиску элемента с заданными свойствами в массиве • Заполнению элементов двумерных массивов значениями в соответствии с заданными правилами • Добавлению/замене элементов в массиве • Исключению / добавлению строк и столбцов со значениями • Симметричным преобразованиям квадратных массивов
<ul style="list-style-type: none"> • Описание понятия строки как структуры данных • Идентификация индивидуальных элементов строки символов 	<p>Строки символов Содержание</p> <ul style="list-style-type: none"> • Строки символов: <ul style="list-style-type: none"> ○ Объявление ○ Присваивание значений ○ Вывод • Операции над строками символов: <ul style="list-style-type: none"> ○ Определение длины, 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Проверке правильности объявления строк символов • Проверке правильности ввода строк символов • Вычислению результатов, отображаемых программами, содержащими строки символов

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> Выбор инструментов обработки строк в зависимости от их свойств Использование строк для хранения данных в программе 	<ul style="list-style-type: none"> Сравнение, Поиск, Слияние, Копирование, Удаление. 	<p>Задачи по:</p> <ul style="list-style-type: none"> Преобразованию строк Моделированию операций на строках Проверке свойств строк символов Поиску последовательностей символов с заданными свойствами и т. д.
Учебная единица 6: Функции		
<ul style="list-style-type: none"> Описание понятия подпрограммы Мотивирование использования подпрограмм Анализ моделей проблем, событий и явлений для разделения на связанные подзадачи 	<p>Подпрограммы Содержание</p> <ul style="list-style-type: none"> Компьютерное решение сложных задач <ul style="list-style-type: none"> Разделение на подпрограммы Корреляция подпрограмм Сборка решений Понятие подпрограммы Типы подпрограмм 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Разделению задач со сложной структурой на элементарные подзадачи Определению зависимости между подзадачами Моделированию передачи данных между подзадачами Классификации подпрограмм
<ul style="list-style-type: none"> Мотивирование использования функций в программе Анализ проблем, событий, явлений и ситуаций для компьютерного моделирования с использованием функций Описание структуры функции C / C++ 	<p>Функции Содержание</p> <ul style="list-style-type: none"> Объявление функций <ul style="list-style-type: none"> Заголовок Описание локальные переменные возвращаемое значение Вызов функции <ul style="list-style-type: none"> параметры возвращение результата Тип void <ul style="list-style-type: none"> Ссылки Параметры со сложной структурой 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Проверке правильности заголовка/вызова функции Оценке результатов, отображаемых программой Контролю значений переменных в процессе пошагового выполнения программы (функция main - функция пользователя) Преобразованию программ с помощью функций <p>Задачи по:</p> <ul style="list-style-type: none"> Расчету простых числовых значений с помощью функций.

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> Разработка собственных функций 		<ul style="list-style-type: none"> Определению эффективности программ, содержащих функции Изменению значений переменных в функциях
<ul style="list-style-type: none"> Описание рекурсивных процессов Идентификация в компьютерных моделях рекурсивных моделей и их компонентов Описание структуры рекурсивной функции Разработка собственных рекурсивных функций 	<p>Рекурсивные функции Содержание</p> <ul style="list-style-type: none"> Рекурсия <ul style="list-style-type: none"> Правило схождения Элементарный случай Рекуррентная формула Рекурсивные функции. <ul style="list-style-type: none"> заголовки структура Контроль выполнения рекурсивных функций. Преобразование функций 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Проверке правильности заголовка / вызова рекурсивной функции Оценке результатов, отображаемых программой, содержащей рекурсивные функции Контролю значениями переменных в рекурсивных процессах Трансформации программ с использованием рекурсивных функций <p>Задачи по:</p> <ul style="list-style-type: none"> Рекурсивным арифметическим расчетам Комбинаторным расчетам Логическим вычислениям Сложным, задачам, сводимым к рекурсивным

Модуль II: Алгоритмы. Введение

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
Учебная единица 1: Алгоритмика		
<ul style="list-style-type: none"> описание концепции алгоритма перечисление свойств алгоритма определение алгоритмов в повседневной человеческой деятельности 	<p>Алгоритмы. История Содержание:</p> <ul style="list-style-type: none"> Понятие алгоритма. история. свойства эпически известные алгоритмы 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Идентификации алгоритмов Описанию свойств алгоритма Объяснению свойств алгоритма Перечислению областей, в которых используются алгоритмы Определению алгоритмов в повседневной деятельности

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> Распознавание графических элементов, используемых для описания алгоритмов логическими схемами реализация в логических схемах алгоритмических действий и операций, описанных на человеческом языке 	<p>Способы описания алгоритмов</p> <p>Содержание:</p> <ul style="list-style-type: none"> Методы представления алгоритмов: <ul style="list-style-type: none"> Человеческий язык, Логические схемы, Псевдокод. Ловушки описания алгоритмов Описание алгоритмов с помощью языков программирования 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Представлению в логических схемах алгоритмов, описанных на человеческом языке Анализу и интерпретации алгоритмов, описанных логическими схемами Разработке программ на языке программирования на основе алгоритмов, описанных логическими схемами Тестированию результатов, полученных при выполнении разработанных программ,
<ul style="list-style-type: none"> Описание алгоритмов решения задач доступными методами Разработка программ, для реализации элементарных алгоритмов 	<p>Элементарные алгоритмы</p> <p>Содержание:</p> <ul style="list-style-type: none"> Наибольший общий делитель (Алгоритм Евклида), Решето Эратосфена (простые числа), Вавилонский алгоритм извлечения квадратного корня, Числа Фибоначчи 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Интерпретации алгоритмов, описанных на человеческом языке Реализации алгоритмов на языке программирования Тестированию корректности реализации алгоритмов <p>Задачи по:</p> <ul style="list-style-type: none"> разрешению ситуаций, в которых используются исследуемые алгоритмы <p>Тематическое исследование:</p> <ul style="list-style-type: none"> Алгоритмы Древней Греции
Учебная единица 2: Алгоритмы на массивах		
<ul style="list-style-type: none"> распознавание основных типов структур данных выбор структур данных, соответствующих решаемым задачам выполнение операций, определенных на структурах данных 	<p>Поиск и подсчет в массивах</p> <p>Содержание:</p> <ul style="list-style-type: none"> Прохождение линейных и двумерных массивов Поиск элементов с заданными свойствами Поиск подпоследовательностей с заданными свойствами Подсчет элементы с заданными свойствами, при прохождении массива Проверка свойств элементов 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Описанию структур данных с заданными свойствами Работе со структурами данных Моделированию прохождения линейных структур данных Моделированию прохождения двумерных структур данных <p>Задачи по:</p> <ul style="list-style-type: none"> определению минимального / максимального значения

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
	<ul style="list-style-type: none"> Приращение и сброс счетчика 	<ul style="list-style-type: none"> Перестановке элементов структуры данных проверке «симметрии» структур подсчету простых / положительных / делящихся на X элементов и т.д.
<ul style="list-style-type: none"> описание процесса сортировки мотивация необходимости использования сортировки в повседневной жизни описание алгоритмов сортировки реализация алгоритма сортировки в задачах программирования 	<p>Упорядочивание элементов в таблице. Пузырьковая сортировка</p> <p>Содержание:</p> <ul style="list-style-type: none"> Роль сортировки в оптимизации алгоритмов Пузырьковая сортировка <ul style="list-style-type: none"> Проверка порядка соседних элементов Перестановка элементов повторения Обратный порядок сортировки 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Проверке наличия порядка элементов структуры данных (таблица, строка) Перестановке элементов структуры данных Упорядочиванию элементов структуры данных <p>Задачи по:</p> <ul style="list-style-type: none"> определению порядка товаров быстрому доступу к элементам структур с экстремальными значениями (первые K наибольшие / наименьшие)
<ul style="list-style-type: none"> описание альтернативных алгоритмов сортировки описание специфики сортировки строк символов описание алгоритма сортировки выбором описание алгоритма сортировки вставкой реализация алгоритмов сортировки в задачах программирования 	<p>Другие алгоритмы сортировки</p> <p>Содержание:</p> <ul style="list-style-type: none"> Сортировка выбором <ul style="list-style-type: none"> Поиск максимального / минимального значения Запоминание индекса Перестановка элементов Сортировка вставкой <ul style="list-style-type: none"> Вставка произвольного элемента в отсортированный подмассив сдвиг элементов Как мы сортируем строки? <ul style="list-style-type: none"> Коды символов лексикографическая сортировка 	<p>Упражнения по:</p> <ul style="list-style-type: none"> моделированию упорядочивания элементов структуры данных (массив, строка) методом выбора перестановке произвольно расположенных элементов в структуре данных Увеличению отсортированной последовательности, добавлением новых элементов. <p>Задачи по:</p> <ul style="list-style-type: none"> Сортировке выбором элементов структуры лексикографической сортировке Сравнению эффективности алгоритмов

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
Учебная единица 3: Методы программирования		
<ul style="list-style-type: none"> • описание принципа решения компьютерных задач методом полного перебора • вычисление множества всех возможных решений задачи программирования • реализация алгоритмов полного перебора для решения задач программирования 	<p>Что означает «полный перебор»?</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Пространство возможных решений для алгоритмов 1D • Пространство возможных решений для алгоритмов 2D • Выбор оптимального решения • Ограничения применения метода 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Поиску и подсчету данных с заданным значением в массивах 1D • Подсчету точек плоскости с заданными свойствами • Генерации всех подмножеств множества <p>Задачи по:</p> <ul style="list-style-type: none"> • Определению оптимального решения дискретных задач: <ul style="list-style-type: none"> ○ Сбалансированных сумм в линейном массиве ○ Задаче о рюкзаке ○ Задаче пересечения кругов
<ul style="list-style-type: none"> • определение рекуррентных формул для элементарных алгоритмов • определение элементарных случаев для рекурсивных вычислений • разработка программ, содержащих рекурсивные подпрограммы 	<p>Рекурсия</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Рекурсия. Элементарные алгоритмы и не только. • Что такое рекурсия <ul style="list-style-type: none"> ○ Сокращение размера задачи ○ Элементарные случаи ○ Правило схождения. • Элементарные рекурсивные алгоритмы <ul style="list-style-type: none"> ○ Расчет степени ○ Факториал ○ Комбинации ○ Вычисление выражений 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Моделированию рекуррентных формул и элементарных случаев • Проверке правила схождения • Моделированию рекурсивных вычислений • Разработка рекурсивных функций <p>Задачи по:</p> <ul style="list-style-type: none"> • Рекурсивному вычислению сумм и произведений • Рекурсивному вычислению элементов известной числовой последовательности • Комбинаторным рекурсивным вычислениям • Рекурсивной обработке последовательностей символов

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> описание волнового алгоритма реализация алгоритма на языке программирования процедурного типа выявление реальных ситуаций, в которых можно использовать алгоритм Ли 	<p>Волновой алгоритм</p> <p>Содержание:</p> <ul style="list-style-type: none"> Алгоритм Ли <ul style="list-style-type: none"> описание моделирование Структуры данных 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Моделированию алгоритма Ли Аналізу принципа распространения волн <p>Задачи по:</p> <ul style="list-style-type: none"> Решению с помощью алгоритма Ли задач на основе повседневных процессов <ul style="list-style-type: none"> задачи кратчайшего пути в лабиринте задачи фотографии
<ul style="list-style-type: none"> описание техники Divide et Impera разработка функций для быстрой сортировки на основе техники Divide et Impera описание задачи Ханойские башни выявление задач, решаемых с помощью техники Divide et Impera 	<p>Рекурсивные методы: Divide et Impera</p> <p>Содержание:</p> <p>Откуда происходит название?</p> <ul style="list-style-type: none"> Рекурсивное моделирование <ul style="list-style-type: none"> Этап разделения Этап сборки Рекурсивные сортировки <ul style="list-style-type: none"> QuickSort MergeSort Замечательные задачи – Ханойские башни <ul style="list-style-type: none"> История задачи Как решить? 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Моделированию сортировки разделением (меньше - больше) Моделированию сортировки разделением (на равные части) Формулирование условий для продолжения / остановки процессов разделения <p>Задачи по:</p> <ul style="list-style-type: none"> Реализации алгоритма быстрой сортировки QuickSort Реализации алгоритма быстрой сортировки MergeSort Сортировке больших объемов данных
<ul style="list-style-type: none"> описание компонентов метода перебора с возвратом определение пространства решений задачи 	<p>Рекурсивные методы: Перебор с возвратом</p> <p>Содержание:</p> <ul style="list-style-type: none"> Перебор с возвратом: <ul style="list-style-type: none"> Компоненты метода «движения расширения» Движения возврата Перебор с возвратом в лабиринте 	<p>Упражнения по:</p> <ul style="list-style-type: none"> Интерактивному моделированию поиска выхода из лабиринта Формулировке условий продолжения и условия возврата для 2 и 3 мерных лабиринтов

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> • Формулировка условий расширения (движения вперед) и возвращения (движения назад) для компьютерных решений задач 	<ul style="list-style-type: none"> ○ Определение всех возможных маршрутов ○ Определение маршрутов, которые избегают предопределенных участков лабиринта • перебор с возвратом на шахматной доске: <ul style="list-style-type: none"> ○ Полное покрытие шахматной доски движениями шахматного коня ○ Проблема N ферзей ○ Аттила и король 	<p>Задачи по:</p> <ul style="list-style-type: none"> • Реализации перебора с возвратом в рекурсивных алгоритмах • Решению задачи лабиринта в стандартных условиях • Решению задачи лабиринта с дополнительными ограничениями • Компьютерному решению задачи полного покрытия шахматной доски движениями коня • Компьютерному решению задачи размещения N ферзей на шахматной доске • Компьютерному решению задачи Аттила и король
Учебная единица 4: Алгоритмы на графах		
<ul style="list-style-type: none"> • определение элементов графа • определение понятий относящихся к графам • определение типа графа • описание графа с использованием различных методов представления • разработка программ перехода между формами представления графа 	<p>Графы. Определения. Способы представления</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Основные понятия: <ul style="list-style-type: none"> ○ Вершина ○ Ребро ○ Путь ○ Цикл • Типы <ul style="list-style-type: none"> ○ Ориентированный граф ○ Неориентированный граф ○ Планарный граф ○ Полный граф • Методы компьютерного представления: <ul style="list-style-type: none"> ○ Матрица смежности ○ Список ребер 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Идентификации элементов графа • Определению типа графа • Формированию структур данных для представления графа <p>Задачи по:</p> <ul style="list-style-type: none"> • Идентификации «центральных» вершин (подключенных непосредственно к наибольшему количеству вершин) • Идентификации изолированных вершин • Преобразованию способа представления графа

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
<ul style="list-style-type: none"> • объяснение понятия поиска вершин в графе • определение понятий компонент связности и сильной связности • выявление ситуаций и проблем, требующих использования алгоритмов поиска для их компьютерного решения. 	<p>Поиск в графе</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Алгоритмы поиска <ul style="list-style-type: none"> ○ Поиск в глубину – DFS ○ Поиск в ширину - BFS. • Связность <ul style="list-style-type: none"> ○ Компоненты связности ○ Компоненты сильной связности 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Поиску в ширину в графе • Поиску в глубину в графе • Идентификации связных компонент в неориентированных графах • Идентификации компонент сильной связности в ориентированных графах <p>Задачи по:</p> <ul style="list-style-type: none"> • Определению наличия пути между заданными вершинами • Поиску компонент связности (задача о политических группировках)
<ul style="list-style-type: none"> • визуальная идентификация графов со свойствами деревьев • описание понятия дерева минимальной стоимости • построение дерева минимальной стоимости 	<p>Остовные деревья минимальной стоимости</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Деревья • Деревья минимальной стоимости. <ul style="list-style-type: none"> ○ Алгоритм Prim, ○ Алгоритм Kruskal 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Интерактивному построению деревьев минимальной стоимости (Kruskal) • Интерактивному построению деревьев минимальной стоимости (Prim) <p>Задачи по:</p> <ul style="list-style-type: none"> • Расчет минимального веса остовного дерева (Kruskal) • Реконструкции железной дороги (Prim)
<ul style="list-style-type: none"> • объяснение понятия кратчайшего пути • описание алгоритмов определения кратчайшего пути • внедрение алгоритмов определения минимального расстояния 	<p>Кратчайшие пути в графе</p> <p>Содержание:</p> <p>Определения</p> <ul style="list-style-type: none"> • Кратчайший путь <ul style="list-style-type: none"> ○ Между двумя вершинами ○ От заданной вершины до всех остальных вершин ○ Между всеми парами вершин • Алгоритмы <ul style="list-style-type: none"> ○ Dijkstra, ○ Floyd 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Определению кратчайшего пути (Dijkstra) интерактивно • Определению кратчайшего пути (Floyd) <p>Задачи по:</p> <ul style="list-style-type: none"> • Генерированию матрицы расстояний

Единицы компетенций	Единицы содержания / ключевые понятия	Рекомендуемая учебная деятельность и школьные продукты
		<ul style="list-style-type: none"> • Восстановлению траектории минимального расстояния • Определению всех минимальных расстояний • Оптимальному расположению
Учебная единица 5: Эвристические алгоритмы		
<ul style="list-style-type: none"> • определение пространства решений задачи • формулировка критерия принятия элементов • установление ограничений • формирование решений • выявление задач, которые могут быть решены с помощью алгоритмов Greedy 	<p>Жадные алгоритмы</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Быстрые решения и точные решениями – кто лучше? <ul style="list-style-type: none"> • Этапы техники Greedy • Пространство решений • Критерий принятия • Коэффициент принятия • Ограничения 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Определению пространства решений • Определению ограниченной задачи • Улучшению критерия принятия <p>Задачи по:</p> <ul style="list-style-type: none"> • Максимизации / минимизации • Моделированию • Графам
<ul style="list-style-type: none"> • Описание игрового поля • Выбор начальных форм • Моделирование решений со специфическими свойствами 	<p>Игра «Жизнь»</p> <p>Содержание:</p> <ul style="list-style-type: none"> • Описание правил • Моделирование состояний • Рекуррентные формулы • Анализ текущего состояния 	<p>Упражнения по:</p> <ul style="list-style-type: none"> • Моделированию этапов игры • Выбору исходной конфигурации • Реализации алгоритма <p>Задачи по:</p> <ul style="list-style-type: none"> • Генерированию фигур <ul style="list-style-type: none"> ○ Статических ○ Периодических ○ Динамических ○ Типа поезд ○ Типа хищник

Рекомендации по оцениванию

Ориентация процесса преподавания-обучения-оценивания на формирование компетенций генерирует непрерывную структуру оценивания.

Оценивание:

- является постоянным процессом, важнейшим аспектом образовательного процесса и эффективной практикой;
- стимулирует обучение, обучение и развитие компетенций;
- ориентируется на необходимость сравнения уровня подготовки учащихся и специфических компетенций, единиц компетенций (подкомпетенций) и (оперативных) целей каждого урока;
- основано на государственных образовательных стандартах - стандартах компетентности - ориентированных на то, что он будет знать; что он будет знать, как делать; и каким будет ученик по завершении курса;
- включает в себя использование различных методов (традиционных и современных);
- является регулирующим процессом, который определяет качество школьной деятельности;
- должно направлять учащихся к надлежащей самооценке и постоянному улучшению успеваемости.

Основные цели оценивания:

- мотивация учащихся и получение обратной связи;
- стимулирование у учащихся усилий по развитию способности к самооценке (формирующая самооценка – это процесс, с помощью которого сам ученик должен судить о качестве своей работы в соответствии с предварительно определенными целями и предлагаемыми критериями оценки);
- формирование навыков взаимной оценки (взаимная оценка представляет собой процесс оценочного взаимодействия, ориентированного на генерацию значимых суждений на основе предопределенных критериев);
- подчёркивание успеха, реализуя таким образом принцип центрирования на личности оцениваемого.

В процессе обучения преподаватель будет выполнять:

1. первоначальное оценивание, для выполнения функции прогнозирования;
2. текущее оценивание, для выполнения обучающей функции;
3. финальное оценивание (суммативное), для выполнения диагностической функции.

Ценность формирующего оценивания заключается в постоянном и непрерывном развитии у учащихся компетенций, отраженных в образовательных стандартах. Задания формирующего оценивания должны быть распределены по степеням сложности, с тем чтобы обеспечить индивидуализацию оценки и дальнейшую мотивацию оцениваемых учеников, что позволит максимально использовать интеллектуальный потенциал каждого учащегося

Разработка заданий для оценивания будет выполняться в контексте и в рамках соответствующих систем классификации.

Итоговое оценивание осуществляется в конце каждого модуля, семестра, учебного года. В качестве составных компонентов инструмента оценивания рекомендуется использование компьютерных тестов и задач для решения с помощью компьютера. В целом, проведенное оценивание покажет если по окончании изучения модуля ученик получает навыки, определенные единицами компетенций, описанными в куррикулуме для данного раздела.

Методы, используемые для непрерывного оценивания, включают устные или письменные опросы, интерактивные методы: тематические исследования, практические работы, проекты, интерактивное компьютерное тестирование.

Для успешного выполнения практической деятельности, учащиеся должны быть заранее проинформированы преподавателем относительно:

- тематики практической работы;
- принципа оценивания (барем / шкала / критерии оценивания);
- условий выполнения данной деятельности.

Непрерывное оценивание позволяет объективно оценить знания и компетенции студентов, а также их прогресс.

Контрольные задания будут содержать упражнения и задачи, посредством которых приоритетно будет производиться оценивание формирования и развития компетенций, а не только отдельно взятые знания и умения.

Посредством дидактических заданий различного уровня сложности учитель направляет и контролирует познавательную деятельность учащихся, подчеркивает, что и как они должны изучить, формируя у обучаемых интеллектуальный стиль работы. Выполненное таким образом оценивание позволяет избежать неожиданных результатов. Оценивание не выполняется с целью «санкционирования», а позволяет выполнить самопроверку полученных результатов, превращая ученика в субъекта собственного обучения.

Совокупность этих методов позволяет учителю получать ориентиры и собирать информацию о учебной деятельности ученика, предоставляя ему возможность продемонстрировать полученные навыки в различных контекстах и ситуациях.

Основываясь на этой информации, учитель объективно формирует свое значимое суждение о приобретенных знаниях и прогрессе учащихся. Оценки, проведенные в конце учебного года, продемонстрируют наличие подкомпетенций, перечисленных в данном куррикулуме.

В контексте принципов приоритетного и доминирующего оценивания в образовательном процессе важно соотношение между текущим и непрерывным оцениванием. Успех урока является результатом достижения поставленных целей. В этом контексте компонента **оценивания** является обязательной для каждого урока, и в рамках именно этой компоненты будет оцениваться уровень достижения целей урока.

Проведенные оценивания в обязательном порядке будут содержать задания, решение которых требует использования междисциплинарных связей. Важно, чтобы каждый учащийся, учитель и родитель/опекун знали, что оценивание при любых обстоятельствах должно быть объективным.

Библиография

Библиографические ресурсы

1. Cerchez Emanuela, Șerban Marin. Limbalul C / C++ pentru liceu (vol I), Polirom, Iași, 2007
2. S. Corlat, A. Gremalschi Grafuri. metodologia predării în cadrul instruirii de performanță la disciplinele Matematică & Informatică Universitatea de Stat din Tiraspol, 2013.
3. Sedgewick, Robert. Algorithms in C++. Parts (I - IV), Fundamentals, Data structures, sorting, searching. Addison – Wesley, 1999
4. Sedgewick, Robert. Algorithms in C++. Part V, Graph algorithms. Addison – Wesley, 1999
5. Ryuhei Uehara First Course in Algorithms Through Puzzles, Springer, 2019

Нормативные документы

1. Cadrul de referință al Curriculumului Național, Chișinău, 2017
2. Repere conceptuale privind elaborarea curriculumului la disciplinele opționale, Chișinău, 2017