



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

*Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз*

Инвестира във вашето бъдеще



Европейски социален фонд

УЧЕБНА ПРОГРАМА ПО ИНФОРМАТИКА ЗА XI КЛАС*

ПРОФИЛИРАНА ПОДГОТОВКА – МОДУЛ 2

* Краен вариант от изпълнение на Дейност 3.3. по ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

*Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз*

Инвестира във вашето бъдеще



Европейски социален фонд

УЧЕБНА ПРОГРАМА ЗА ПРОФИЛИРАНА ПОДГОТОВКА ПО ИНФОРМАТИКА

МОДУЛ 2 „СТРУКТУРИ ОТ ДАННИ И АЛГОРИТМИ”

КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА НА МОДУЛА

Обучението в Модул 2 „Структури от данни и алгоритми” е насочено към овладяване на базисни знания, умения и отношения, свързани с проектирането, създаването и прилагането на разнообразни структури от данни и алгоритми над тях и с изграждането на компетентности на ученика за моделиране, организация и управление на информацията, а така също познавателни и социални компетентности. Учебната програма предполага развиване на умения за учене през целия живот: застъпени са предимно практически ситуации, в които ученикът трябва да заеме активна позиция – да изследва, да предлага, да сравнява, да комбинира различни алгоритми и структури от данни, да прилага придобити вече компетентности. Всяка тема завършва с реализирането на конзолни и графични приложения.

В тази връзка учебната програма (а и обучението като цяло) по информатиката в **Модул 2 “Структури от данни и алгоритми”** за 11. клас е съобразена със система от взаимносвързани фактори, по-важните от които са:

1. Мястото на предмета в учебния план:
 - брой на часовете;
 - връзките и значението му (в хоризонтален и вертикален план) с другите модули и учебни предмети, особено с математика и информационни технологии (ИТ).
2. Държавните образователни изисквания (ДОИ) по информатика.
3. Съвременното състояние, понятийния апарат, логическата структура, методите и средствата на научната област.
4. Технологичното оборудване в системата на средното образование – компютри, системен и приложен софтуер, мрежова и комуникационна инфраструктура, мултимедийни и други съвременни дидактически средства.
5. Възрастовите (познавателни и психологични) възможности и интереси на неголяма, специфична извадка от учениците, които са насочили своето образование в областта на природо-математическите дисциплини.



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

6. Процесът на обучението по информатика притежава потенциални възможности за личностно изграждане и развитие на ученика – формиране на абстрактно и логическо мислене, възпитание и формиране на адекватно отношение към заобикалящата действителност.
 7. Учебният предмет е и възлов инструмент, с помощта на който могат да се мотивират, ориентират, привличат и развият младите хора, така че впоследствие да бъдат създадени специалисти, способни да осигурят кадрово информационното обслужване на обществените, стопанските и научните обществени сфери.
 8. Учебната програма е предназначена за профилирана подготовка във втори гимназиален етап. Съдържанието е предвидено да се реализира в рамките на 72 учебни часа през втория учебен срок на 11. клас. Предназначена е за ученици в математическите гимназии на Република България. Разработена е в съответствие с цитираните по-горе система от фактори (1 - 7) и определя:
- очакваните резултати от обучението в Модул 2 за 11. клас в съответствие с ДОО;
 - обема и структурата на учебното съдържание, представени в примерен тематичен план;
 - специфичните изисквания за провеждане на обучението;
 - препоръчителното разпределение на часовете;
 - формите и методите за оценяване.

УЧЕБНО СЪДЪРЖАНИЕ И ОЧАКВАНИ РЕЗУЛТАТИ ПО ТЕМИ

Теми	Очаквани резултати	Нови понятия
1. Алгоритми и методи на класове		
1.1. Реализация на алгоритми чрез методи	<ul style="list-style-type: none"> • Описва методи, реализиращи алгоритъм за специфично поведение на обект от даден клас • Разбира технологията на обръщение към “предефиниран” (<i>override</i>) метод • Разбира разликите между “презареждане” (<i>overload</i>) и “предефиниране” (<i>override</i>) на методи 	<ul style="list-style-type: none"> • Предефиниране (<i>override</i>) на метод
1.2. Рекурсия	<ul style="list-style-type: none"> • Разграничава основни видове рекурсия 	<ul style="list-style-type: none"> • Рекурсия



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	<ul style="list-style-type: none"> • Намира рекурентни зависимости • Разбира програмната техника рекурсия • Формулира гранични условия на рекурсия • Описва рекурсивни методи • Описва рекурсивни методи итеративно • Описва итеративни методи рекурсивно 	<ul style="list-style-type: none"> • Пряка и косвена рекурсия • Гранично условие • Рекурсивен метод
2. Класификация на структурите от данни		
2.1. Абстрактни типове от данни	<ul style="list-style-type: none"> • Обяснява понятието абстрактен тип данни • Класифицира основните абстрактни типове данни • Познава колекции, реализиращи абстрактни типове данни 	<ul style="list-style-type: none"> • Абстрактен тип данни (АТД) • Колекция
2.2. Класификация на структурите от данни	<ul style="list-style-type: none"> • Изброява основни структури от данни: масив, символен низ, линеен списък (едносвързан, двусвързан, цикличен), стек, опашка, граф, кореново дърво • Познава структурата, характерните методи и приложения на линейни абстрактни типове данни • Познава структурата, характерните методи и приложения на речници, хеш-таблици и множества • Познава нелинейни (разклонени) АТД • Познава структурата, характерните методи и приложения на граф • Познава структурата, характерните методи и приложения на дърво 	<ul style="list-style-type: none"> • Линейни АТД • Разклонени АТД • Речник • Хеш таблица • Множество • Граф • Дърво
3. Символен низ		
3.1. Символен низ	<ul style="list-style-type: none"> • Познава стандартен клас, реализиращ символен низ • Декларира, създава и инициализира символен низ 	<ul style="list-style-type: none"> • Символен низ • Празен низ • Дължина на низ
3.2. Обработка на символни низове	<ul style="list-style-type: none"> • Познава вградени методи за обработка на символен низ • Сравнява низове лексикографски 	<ul style="list-style-type: none"> • Лексикографско сравнение • Конкатенация



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	<ul style="list-style-type: none"> Използва символни низове за обработка на текст: <ul style="list-style-type: none"> - изтрива подниз - вмъква подниз - заменя един подниз с друг Търси подниз в низ по пълно съвпадение: <ul style="list-style-type: none"> - проверява дали даден подниз се среща в даден низ - намира първо срещане на подниз в низ - намира последно срещане на подниз в низ - намира всички срещания на подниз в низ 	<ul style="list-style-type: none"> Подниз
3.3. Регулярни изрази	<ul style="list-style-type: none"> Познава основни метасимволи и техните предназначения Познава метасимволи за групиране и количество Разпознава дали даден израз отговаря на определени синтактични правила Проверява със средствата на езика дали даден символен низ отговаря на зададен шаблон Търси подниз в низ по шаблон Валидира данни с помощта на регулярни изрази 	<ul style="list-style-type: none"> Метасимвол Регулярен израз
4. Едномерен масив		
4.1. Едномерен масив	<ul style="list-style-type: none"> Декларира и инициализира числови масиви с инициализиращ списък и случайни числа Извлича текущия размер на масив Декларира и обработва масив от низове Дефинира масив от обекти Инициализира масив от обекти Създава метод за извеждане на елементите на масив в символен низ Променя размера на масив по време на изпълнение на програмата 	Размер на масив



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

4.2. Сортиране на масив	<ul style="list-style-type: none"> Разпознава сортирана редица Познава алгоритмите за сортиране – „метод на мехурчето“ и „метод на пряката селекция“ Използва средствата на езика за предефиниране на метод за сравнение на обекти от даден клас Реализира алгоритми за сортиране – метод на мехурчето и метод на пряката селекция Използва методи на стандартни библиотеки за ефективно сортиране на масив 	<ul style="list-style-type: none"> Сортиране на масив Възходящо/низходящо сортиране
4.3. Техники за работа с сортиран масив	<ul style="list-style-type: none"> Прилага алгоритъма за двоично търсене в сортиран масив Обоснова избиране на метод на търсене в практически задачи Използва методи на класове от стандартни библиотеки за търсене на елементи по дадени критерии Прилага алгоритъм за сливане на сортирани редици Описва метод за сливане на сортирани масиви 	<ul style="list-style-type: none"> Двоично търсене Сливане на сортирани редици
4.4. Комбинаторни алгоритми	<ul style="list-style-type: none"> Генерира комбинаторни конфигурации със и без повторение Прилага комбинаторни алгоритми в игрови ситуации Моделира и реализира решение на проблем с помощта на комбинаторни алгоритми 	<ul style="list-style-type: none"> Комбинаторни конфигурации Вариации Пермутации Комбинации
5. Многомерен масив		
5.1. Многомерен масив	<ul style="list-style-type: none"> Осмисля структурата на многомерен масив Изброява възможни приложения на многомерен масив 	<ul style="list-style-type: none"> Многомерен масив
5.2. Двумерен масив	<ul style="list-style-type: none"> Дефинира двумерен масив от обекти Създава и инициализира двумерен масив Представя таблични данни с помощта на двумерен масив Обработва таблични данни с помощта на двумерен масив Обхожда двумерен масив по редове, стълбове и диагонали 	<ul style="list-style-type: none"> Двумерен масив Ред Стълб Диагонал



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	<ul style="list-style-type: none"> • Използва двумерен масив при моделиране на решение на проблем 	
6. Линеини структури от данни		
6.1. Линеини структури от данни	<ul style="list-style-type: none"> • Познава общите характеристики на линеини структури от данни • Познава колекции, реализиращи линеини структури от данни 	<ul style="list-style-type: none"> • Списък • Стек • Опашка
6.2. Списък	<ul style="list-style-type: none"> • Разбира структурата от данни вектор • Разбира структурата от данни линеен едно свързан списък • Използва реализация на списък от стандартните библиотеки • Добавя елемент в списък • Обхожда линеен списък • Търси елемент в списък • Извлича и използва елемент от списък • Премахва елемент от списък • Използва списък за решаване на практически проблеми 	<ul style="list-style-type: none"> • Итератор
6.3. Стек	<ul style="list-style-type: none"> • Разбира структурата от данни стек • Използва реализация на стек от стандартните библиотеки • Добавя елемент в стек • Премахва елемент от стек • Осъществява достъп до върха на стек • Използва стек за решаване на практически проблеми 	<ul style="list-style-type: none"> • Върх на стека
6.4. Опашка	<ul style="list-style-type: none"> • Разбира структурата от данни опашка • Използва вградена реализация на опашка • Добавя елемент в опашка • Премахва елемент от опашка • Използва опашка за решаване на практически проблеми 	<ul style="list-style-type: none"> • Начало на опашка • Край на опашка
7. Структури от данни –		



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

съпоставка и препоръки		
7.1. Сложност на алгоритъм	<ul style="list-style-type: none"> Знае как да оценява сложността на алгоритъм <ul style="list-style-type: none"> сложност по време сложност по памет сложност по среден брой операции Посочва примери за алгоритми с различна сложност при решаване на един и същи проблем Посочва примери за алгоритми с пълно изчерпване 	<ul style="list-style-type: none"> Сложност на алгоритъм Сложност по време Сложност по памет Сложност по среден брой операции
7.2. Сравнение на основните структури	<ul style="list-style-type: none"> Сравнява структурите по време за: <ul style="list-style-type: none"> добавяне на елемент търсене на елемент изтриване на елемент достъп до отделен елемент Сравнява структурите по ефективно заемане на оперативна памет Посочва примери за избор на подходяща структура от данни в зависимост от позволените методи за добавяне и извличане на елементи и характера на моделирания проблем 	<ul style="list-style-type: none"> Константна сложност Логаритмична сложност Линейна сложност Полиномиална сложност Експоненциална сложност
7.3. Структури от параметризирани данни	<ul style="list-style-type: none"> Използва стандартни библиотеки от шаблонни класове моделиращи структури от данни (масив, списък, стек, опашка) Избира параметри за тип с класове и интерфейси при решаване на конкретен проблем 	<ul style="list-style-type: none"> Шаблонни класове (generics) Параметър за тип
8. Файлове и потоци от данни		
8.1. Потоци	<ul style="list-style-type: none"> Знае какво представляват потоците и тяхното предназначение Познава видовете потоци в зависимост от посоката на пренасяните данни и класовете, които ги моделират Познава видовете потоци в зависимост от типа на пренасяните данни Умее да осъществява достъп до данните в потоците 	<ul style="list-style-type: none"> Поток Входен поток Изходен поток Двоичен поток Текстов поток



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	<ul style="list-style-type: none"> Умее да форматира подходящо изходните данни 	
8.2. Методи за работа с потоци	<ul style="list-style-type: none"> Обяснява стандартните класове за работа с потоци Прилага методи за обработка на поток: “създаване”; “отваряне”; “затваряне” 	<ul style="list-style-type: none"> Създаване Отваряне Затваряне
8.3. Текстови файлове	<ul style="list-style-type: none"> Познава и използва различни видове кодиране на текстови файлове Владее стандартни методи за работа с текстови файлове: “четене”; “запис”; “добавяне”; “търсене” Използва стандартни класове за работа с текстови потоци 	<ul style="list-style-type: none"> Четене Запис Добавяне Търсене
8.4. Сериализация на данни	<ul style="list-style-type: none"> Прилага методи за обработка на двоичен файл Знае необходимостта от съхранение и пренасяне на обекти Използва сериализация за запис във файл на инстанции на даден клас Демонстрира десериализация при четене от файл на инстанции на даден клас 	<ul style="list-style-type: none"> Сериализация Десериализация
8.5. Приложение	<ul style="list-style-type: none"> Въвежда, обработва и извежда елементите на масив от/във файлов поток Работа със стандартни диалози за отваряне на файл, запазване на файл и преглеждане на директория 	
9. Обработка на изключения		
9.1. Изключения	<ul style="list-style-type: none"> Знае какво е изключение Познава видове изключения в език за ООП Познава изключения, които могат да бъдат породени при стандартни събития, предизвикани от мишка или клавиатура Разбира предимствата на механизма за прихващане и обработка на изключения Познава видовете изключения в език за ООП Различава основните конструкции за прихващане на изключения 	<ul style="list-style-type: none"> Изключение



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

9.2. Обработка на грешки	<ul style="list-style-type: none"> Обяснява фрагмент от програмен текст с наличие на изключения Умее да прихваща и обработва изключения със средствата на обектно-ориентиран език за програмиране Познава възможността за предизвикване на изключения Посочва добри практики при обработка на грешки 	<ul style="list-style-type: none"> Прихващане на изключение Предизвикване на изключение Обработка на изключение
9.3. Приложения	<ul style="list-style-type: none"> Прилага обработка на изключения при валидиране на потребителски вход. Прилага регулярни изрази при обработка на изключения при валидиране на данни 	<ul style="list-style-type: none"> Валидиране на данни
10. Качествен програмен код		
10.1. Именуване на елементите от програмата	<ul style="list-style-type: none"> Знае какво е качествен програмен код Обяснява как да се именуват идентификаторите на: <ul style="list-style-type: none"> класове, интерфейси и методи параметри на методи, променливи и константи 	<ul style="list-style-type: none"> Външно качество (през призмата на потребителя) Вътрешно качество (вътрешна организация)
10.2. Правила за форматиране и подреждане на кода	<ul style="list-style-type: none"> Разбира форматирането на кода като подобряване на неговата четивност чрез разкриване на логическата му структура Прилага правилата за форматиране на метод Спазва правилата за подредба на съдържанието на класа Обяснява правилата за форматиране на цикли и условни конструкции Владее правилата за пренасяне и подравняване 	<ul style="list-style-type: none"> Автоматично форматиране на кода Код-конвенции за форматиране
10.3. Добри практики за изграждане на висококачествени методи	<ul style="list-style-type: none"> Обяснява принципите на обектно-ориентираното програмиране Обосновава необходимостта от писане на методи Подобрява четивността и прегледността на кода, отделяйки всяка логически обособена функционалност, в метод Спазва общоприети норми и конвенции за именуване на методи и съставяне на програмен текст 	<ul style="list-style-type: none"> Обхват, живот, активност на променлива Израз с една операция Контрол на входните данни



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	<ul style="list-style-type: none"> • Разбира понятията “обхват, живот, активност” на променлива • Правилно използва изрази • Прави проверка за коректност на входните данни • Изследва ефективността на методи с еднаква функционалност в различни реализации 	
10.4. Принципи за качествена документация на кода	<ul style="list-style-type: none"> • Създава самодокументиращ се код • Вмъква подходящи коментари • Преработва код 	<ul style="list-style-type: none"> • Самодокументиращ се код • Преработка на код (refactoring)
10.5. Верификация и валидация на цялостно решение	<ul style="list-style-type: none"> • Тества функционалността на отделните компоненти от кода (Unit Testing) • Тества интеграцията между компонентите (Integration Testing) • Тества функционалността на цялостното приложение (System Testing) • Проверява дали са удовлетворени изискванията на потребителя (Acceptance Testing) 	<ul style="list-style-type: none"> • Black box testing • White box testing • Gray box testing

ХОРАРИУМ ЗА МОДУЛА

Годишен брой часове за изучаване на модула в 11. клас – 72 часа

Допълнителни уточнения за конкретния модул.

- Програмата предвижда обучението да се извършва на базата на език за визуално програмиране – Microsoft C#.NET) или Java по избор на преподавателя.
- Препоръчват се следните среди за програмиране: актуални версии на Microsoft Visual Studio (професионална или свободноразпространяема версия (за C#.NET), Netbeans или Eclipse (за Java).
- Препоръчва се обучението да се провежда предимно под формата на комбинирани уроци.



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

- Препоръчва се занятията да се провеждат в блок от два учебни часа и всеки ученик да разполага със собствено работно място за работа в съответната програмна среда.
- Препоръчва се по възможност на учениците да се предостави свободен достъп до компютърните зали извън редовните часове за упражнения и работа по проекти.

Препоръчително разпределение на часовете:

За нови знания	до 24 часа	до 40 %
За преговор	до 4 часа	
За обобщение	до 4 часа	
Практически дейности/лабораторни упражнения	до 34 часа	до 60%
За контрол и оценка (за входно и изходно ниво, текуща проверка и контролни работи)	до 6 часа	

ОЦЕНЯВАНЕ

Форми на оценяване:

- Устна

Устната форма на оценка е удачна при групови обсъждания, дискусии, генериране на идеи. Оценяват се мненията и аргументите на ученика. Не се препоръчва устно изпитване, отнасящо се до запаметяване на теория.

- Писмена

Писмената форма е подходяща при проверка и оценка на теоретични знания и разбиране на синтаксиса и семантиката на даден оператор. Би могла да включва въпроси с избираем отговор, задачи за вида на използваната структура от данни, задачи за проследяване на работата на алгоритъм или програмен фрагмент, задачи за откриване на логически грешки в алгоритъм или програмен фрагмент. Препоръчва се писменото изпитване да бъде във формата на ДЗИ.

- Практическа

Тъй като предметът е с предимно практическа насоченост, препоръчва се това да е преобладаващата форма в оценяването на постиженията на учениците. Практическо изпитване се прави на базата на портфолио, съдържащо резултати от практическата работа в клас, защитена домашна работа и разработка на проект в качеството му на изходно ниво от обучението. Желателно е оценката да отразява всички етапи от практическата



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

разработка (анализ, моделиране, програмиране, документация, защита и др. според заданието), като за тази цел учителят трябва да подготви съответните критериални матрици.

Съотношение при формиране на срочна и годишна оценка:

Текущи оценки от работа в клас, участие в групови обсъждания и дискусии	10%
Текущи оценки от домашни работи	20%
Текущи оценки от практически задания в клас	30%
Оценка на изходно ниво	20%
Оценки от контролни работи	20%

ДЕЙНОСТИ И МЕЖДУПРЕДМЕТНИ ВРЪЗКИ

Дейност 1.	Самостоятелно търсене и представяне в ефективна форма допълнителна информация, свързана с изучаваните теми. Учениците развиват умения за четене, социални и граждански компетентности, създават се интердисциплинарни връзки с българския език, упражняват самоконтрол при изпълнение на различни практически задачи.
Дейност 2.	Структурира информация във вид на данни в контекста на конкретната информационна задача, моделира реални жизнени процеси. На учениците се дава възможност да демонстрират социални и граждански компетентности – да дискутират и оценяват различни идеи, да работят в екип, да изразяват мнение, да отстояват собствена позиция. Развива ключови компетентности в областта на българския език и чуждите езици – писмено и устно изразяване.
Дейност 3.	Идентифициране и дефиниране на стандартите в информатиката и използване на конвенция за именуване. Учениците демонстрират дигитални компетентности и умения за общуване на чужди езици. Осъществява се интердисциплинарна връзка с предмети, развиващи граждански компетентности - история и цивилизация,



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	география и икономика, предметите от философския цикъл.
Дейност 4.	Самостоятелно търсене и представяне в ефективна форма на информация, свързана с приложението на езици за програмиране за автоматизиране на информационните дейности в ежедневието. Класифициране на информацията според основните групи езици за програмиране. Изготвяне на материали и представяне пред публика. На учениците се дава възможност да демонстрират дигитални компетентности, инициативност и предприемчивост. Междупредметни връзки могат да се осъществят с български език и литература и всички изучавани дисциплини в зависимост от избраните от учениците теми.
Дейност 5.	Създаване на програма за пакетиране на стоки. Самостоятелна реализация на алгоритъма на Евклид. Учениците визуализират практическата му приложимост, демонстрират компетентности инициативност и предприемчивост.
Дейност 6.	Преобразуване на запис на число от десетична в р-ична бройна система. Учениците развиват дигитални и математически компетентности, демонстрират обществени и граждански компетентности за кодиране на информация.
Дейност 7.	Представяне на рекурентни редици. Редактират, компилират и изпълняват готова програма, създадена за различни сфери на обществения живот. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности. Осъществяват се междупредметни връзки с биология, химия, физика, информационни технологии и математика.
Дейност 8.	Визуализират фрактална графика. Осъществяват междупредметна връзка с Модул 1 и физика и астрономия. Учениците демонстрират математически и дигитални компетентности в областта на природните науки и на технологиите.
Дейност 9.	Реализират визуално игра за двама. Учениците развиват дигитални, математически, социални и граждански компетентности. Осъществяват се междупредметни връзки с информационни технологии, математика и предметите от философския цикъл.
Дейност 10.	Разделяне на всички думи в едно изречение. Описват алгоритми за търсене в низ, прилагат граматически правила в българския език за конструиране на изречение. Учениците развиват дигитални, математически компетентности



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	и основни компетентности в областта на родния език.
Дейност 11.	Проверка за валидност на дата, електронен адрес, телефонен номер. Въвеждане и извеждане на текстови данни в приложения, свързани с бита и бизнеса. Учениците развиват дигитални и основни компетентности в областта на технологиите, социални и граждански компетентности. Осъществяват се междупредметни връзки с биология, химия, физика, информационни технологии и математика.
Дейност 12.	Сортират лексикографски множество от обекти по текстови компоненти. Прилагат граматически правила в българския език. Осъществяват междупредметна връзка с български език и литература и природни науки.
Дейност 13.	Организируют речник. Развиват дигитални компетентности, ключови компетентности за общуване на роден и чужд език. Осъществяват междупредметни връзки с математика, физика, химия, биология, география и икономика, история и цивилизация.
Дейност 14.	Коригиране на файл със субтитри. Работа в интегрирана среда на визуално програмиране – обектно-ориентиран език. Моделиране на реален процес. Учениците демонстрират компетентности в областта на математиката, природните науки и дигитални такива. Осъществяват се междупредметни връзки с информационни технологии, математика и природонаучни дисциплини.
Дейност 15.	Калкулатор на данъци. Използване средствата от средата за програмиране при установяване и отстраняване на грешки. Създаване на програми с дружелюбен графичен интерфейс. Проучване, обобщаване и представяне на информация за грешки в програмата. Развиват математически и дигитални компетентности за анализ на различни ситуации и умения за вземане на решение. Използване на информация от интернет на български и чужд език.
Дейност 16.	Моделира решаване на статистически проблеми - средноаритметично, медиана, мода и др. Реализират междупредметни връзки с природонаучните дисциплини - физика, химия, биология, география и история.
Дейност 17.	Представяне и изчертаване на многоъгълник. Намиране лице на многоъгълник. Осъществяват междупредметна връзка с математиката – създаване на програма за визуализация на фигури, свързани с различни изчисления. Учениците развиват обществени и граждански компетентности.
Дейност 18.	Моделиране на стрелба по мишена и оценяване по различни видове спортни състезания – биатлон, волейбол, ски-



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз



Европейски социален фонд

Инвестира във вашето бъдеще

	слалом, баскетбол, гимнастика и т.н. Визуализиране. Осъществява се междупредметна връзка с математика и физическо възпитание и спорт. Учениците развиват дигитални и математически компетентности.
Дейност 19.	Визуализират синоптична прогноза, използвайки масив от стойности. Изработване и анализ на хистограма за средната температура в определен район на изследване. Статистика на най-високи/най-ниски температури. Осъществява се междупредметна връзка с математика и география и икономика. Създаване на обществени и граждански компетентности.
Дейност 20.	Моделиране на полином и действия с него. Осъществява се междупредметна връзка с математика – реализиране на алгоритъм за стойност на полином по схемата на Хорнер, намиране на сума и произведение на полиноми, с физика и математика – намиране корените на полином.
Дейност 21.	Събиране на информация от различни източници. Визуализиране на информация за 10-те града с най-голямо замърсяване на въздуха. Трансформиране на информацията във формат на данни. Сортиране на стойностите в структура масив. Търсене на град със замърсяване на въздуха над средното за Европа. Осъществява се междупредметна връзка с химия и опазване на околната среда, с география и икономика. Учениците развиват дигитални и математически компетентности.
Дейност 22.	Създаване на модели на задачи от областта на математиката, физиката и реални житейски ситуации, чието решаване изисква използване на комбинаторни алгоритми. Триъгълник на Паскал. Търсене на път в лабиринт. Симулация на тото 2, евроспорт. Реализация и визуализация на морски шах.
Дейност 23.	Осъществява се междупредметна връзка с математика – създаване на списък за визуализация на простите числа в даден интервал, използване на стек за проверка за съответстващи скоби. Учениците развиват обществени и граждански компетентности - създаване и обработка на библиотечен каталог, стеков калкулатор.
Дейност 24.	Интерпретиране на данни от заобикалящите ни процеси и явления. Примерни приложения – визуализация на железопътен сортировъчен възел, обработка на опашка от клиенти за плащане на електроенергия.
Дейност 25.	Създаване на софтуерно приложение, което чете и записва данни в текстов файл. Подреждане на телефонен указател. Търсене в телефонен указател. Търсене в разписанието на конгресна зала за свободен час. Учениците



Европейски съюз

ПРОЕКТ BG051PO001-3.1.04 „Подобряване на качеството на общото образование”

„За по-качествено образование“

*Проектът се осъществява с финансовата подкрепа на
Оперативна програма „Развитие на човешките ресурси”,
съфинансирана от Европейския социален фонд на Европейския съюз*



Европейски социален фонд

Инвестира във вашето бъдеще

	оценяват различни структури от данни, дискутират алгоритмите, анализират използваните класове и връзките между тях. Развиват обществени и граждански компетентности.
Дейност 26.	Създаване на самостоятелно софтуерно приложение. Учениците разработват проект в екип по двама. Развиват се граждански компетентности - форми на поведение за успешно участие в социалния живот, разрешаване на конфликти и проблеми, умения за самостоятелно учене и събиране на информация. Разработват се общественополезни теми.