Приложение № 81 към т. 81

УЧЕБНА ПРОГРАМА ПО ИНФОРМАТИКА
(ПРОФИЛИРАНА ПОДГОТОВКА)

МОДУЛ 1. „ОБЕКТНО ОРИЕНТИРАНО ПРОЕКТИРАНЕ И ПРОГРАМИРАНЕ“

КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА

Информатиката е наука, която се занимава с методите за структуриране, събиране, обработка и разпространение на данни. Постиженията на тази научна област са пряко и динамично свързани със създаването и развитието на високотехнологични средства (компютри, операционни и комуникационни системи, потребителски софтуер, софтуер за разработка на приложения и др.), които са важен инструмент и/или инфраструктура, върху която функционират почти всички сфери на съвременното общество.

В тази връзка, учебната програма (а и обучението като цяло) по информатиката в **Модул 1. „Обектно ориентирано проектиране и програмиране“** за XI 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. Държавният образователен стандарт (ДОС) - изискванията 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 учениците, които са насочили своето образование в областта на природо-математическите дисциплини.
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 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 в първи учебен срок на XI клас. Разработена е в съответствие с цитираните по-горе система от фактори (1 - 7) и определя:
    - очакваните резултати от обучението в Модул 1. за XI клас в съответствие с ДОС;
    - обема и структурата на учебното съдържание, представени в примерен тематичен план;
    - специфични изисквания за провеждане на обучението;
    - препоръчително разпределение на часовете;
    - форми и методи за оценяване.

УЧЕБНО СЪДЪРЖАНИЕ

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
1. Въведение в информатиката		
1.1. *Езици и* ***среди за програмиране***	• Описва основните групи езици за програмиране и обяснява предназначението им	• Стандартен вход • Стандартен изход

	• Разбира предназначението на интегрирана среда за програмиране • Разпознава видове езици за програмиране според тяхното ниво • Разграничава независим от зависим от средата и/или платформата програмен код	• Типизирани и нетипизирани езици за програмиране • Език за структурно програмиране • Език за обектно-ориентирано програмиране • Език за функционално програмиране • Език за логическо програмиране • Събитийноориентирано програмиране • Машинен език • Асемблерни езици • Езици от високо ниво • Транслатор • Компилатор • Интерпретатор
1.2 *Обектно ориентиран подход*	• Дефинира понятието клас от гледна точка на обектно ориентирания подход. • Дава примери на ситуации от реалния свят с използване на обектно ориентирания подход • Разбира различните нива на абстракция при проектиране на клас – взаимозависимост и свързаност, достатъчност, пълнота и простота	• Обектно ориентиран подход • Ниво на абстракция

	• Обяснява връзката между класове чрез примери • Разграничава състояние и поведение на обектите на клас	
2. Класове и обекти		
2.1 *Интегрирана среда за обектно ориентирано програмиране (ООП).*	• Посочва основните компоненти на интегрирана среда за програмиране • Посочва компонентите на проект в интегрирана среда за програмиране • Използва интегрирана среда за програмиране за разработка на софтуерен (програмен) проект • Познава процеса на преобразуване на изходен код на програмен проект до изпълнима програма	• Проект • Специализиран текстов редактор • Свързващ редактор • Редактор за откриване и отстраняване на грешки • Инструмент за преобразуване до изпълним код • Редактор за проектиране на дизайн на графичен потребителски интерфейс
2.2 *Основни елементи на език за обектно ориентирано програмиране*	• Различава ключови, стандартни и потребителски думи • Изброява ключови думи и основни елементи на обектно ориентиран език за програмиране • Използва конвенция за именуване на клас, метод и променливи (стил „камила“ (CamelCasing) и „Паскал“ (PascalCasing)) • Дефинира понятията клас, обект, член-променливи на клас, член-функции (методи) на клас в обектно ориентиран език за програмиране • Различава настолни приложения с графичен и конзолен интерфейс, уеб	• Идентификатор • Ключова дума • Стандартна дума • Потребителска дума • Конвенция за имената • Клас • Член-променлива • Член-функция (метод)

	приложения, приложения за мобилни устройства и вградени системи	• Обект • Инстанция на клас • Данни
2.3 *Обекти*	• Декларира обект от съществуващ клас • Дефинира понятието „конструктор“ • Разбира предназначението на конструктор на клас • Създава обект от съществуващ клас. • Използва референция за достъп до елементи (членове) на текущия обект. • Използва помощни средства на конкретна среда за програмиране • Използва подходящи коментари • Използва инструменти на средата за създаване на автоматична документация	• Конструктор • Референция за достъп до член на клас • Коментар
3. Членове на клас		
3.1 *Член-променливи (свойства) на клас*	• Разпознава членовете на класа, представени чрез UML диаграма на клас • Декларира свойства на клас • Описва синтаксис и семантика на методите за достъп до свойства и за промяна на данни • Генерира декларации на свойства със средствата на средата за програмиране • Разбира предназначението и описанието на стандартни методи за достъп и промяна на стойности на член-променливи	• UML диаграма на клас • Декларация на свойства в клас
3.2 *Методи на клас*	• Разбира предназначението на формалните параметри • Подбира формални параметри на метод • Извиква метод като съпоставя на списъка с формални параметри съответен списък с фактически параметри	• Параметри на метод • Формални параметри • Фактически параметри • Параметри, предавани по

	• Различава технологията на предаване на параметри по стойност и чрез референция • Разпознава връщащи и невръщащи стойност методи • Описва метод, който не връща резултат • Описва метод, връщащ резултат • Задава коректно обръщение към метод • Декларира локални променливи • Посочва област на действие на променливи във фрагменти от програмен код	стойност • Параметри, предавани чрез рефенция • Обръщение към метод • Локални променливи • Област на действие на променливите
3.3 *Стандартен метод на клас*	• Използва стандартен метод на клас, връщащ форматиран текст с текущите стойности на данните на инстанцията • Използва диалогови прозорци за въвеждане и извеждане на данни	
4. Създаване на графичен интерфейс		
4.1 *Графични средства на интегрирана среда за създаване на графичен интерфейс*	• Използва интегрирана среда за програмиране за създаване на графичен интерфейс • Посочва свойствата на елементи на графичен прозорец. • Създава елементи на графичен прозорец (етикет, текстово поле, бутон) • Задава настройки на свойствата на елементи на графичен прозорец • Описва добри практики за именуване на идентификаторите на графични елементи • Открива грешки при именуване на графични елементи	• Модифицирана унгарска нотация за именуване

4.2 *Създаване на елементарен графичен интерфейс*	• Реализира програмно достъп до и промяна на свойството „*текст*“ на графичните елементи • Реализира програмно обработката на събитие по подразбиране при натискане на бутон.	• Обработка на събитие по подразбиране
5. Основни оператори		
5.1 *Аритметични и логически оператори, релации за сравнение, оператор за присвояване и конкатенация*	• Описва синтаксиса, семантиката и приоритета на основните оператори в конкретен език за програмиране • Разбира асоциативността на оператори с равен приоритет • Разбира спецификата на целочислената аритметика и аритметиката в смесени изрази. • Разбира семантиката на префиксен, инфиксен и постфикен запис на оператор • Правилно прилага аритметични и логически операции, както и релации за сравнение в сложни изрази. • Присвоява стойност на променлива • Разпознава случаи на неявно преобразуване на типове данни. • Използва явно преобразуване на типове данни. • Конкатенира текстови данни.	• Унарни и бинарни оператори • Префиксен, инфиксен и постфиксен запис на оператор
5.2 *Оценяване на логически изрази*	• Разбира пълно оценяване на логически израз • Разбира частично (съкратено) оценяване на логически израз	
6. Конструкции за управление на програмата		
6.1 *Алгоритми.*	• Описва основните видове изчислителни процеси – линеен, разклонен и	• UML диаграма на дейност

	цикличен • Разпознава основните видове изчислителни процеси, описани с UML диаграма на дейност • Обосновава необходимостта от прилагане на методи за избягване на дублиране на код • Прилага подхода „*отгоре-надолу“* (top-down) при решаване на задачи	•
6.2 *Линейни изчислителни процеси*	• Разбира последователността на изпълнение на линеен изчислителен процес. • Описва дизайн на линеен алгоритъм чрез UML диаграма на дейност. • Описва алгоритми, които се реализират чрез линейни изчислителни процеси. • Проследява изпълнението на линеен изчислителен процес	• Линеен изчислителен процес
6.3 *Разклонени изчислителни процеси*	• Разбира последователността на изпълнение на разклонен изчислителен процес. • Описва синтаксиса и семантиката на условен оператор „*if“* • Разбира семантиката на вложени условни оператори • Знае и може да прилага законите на Де Морган и закона за двойно отрицание над логически изрази • Описва ефективно вложени условни оператори • Описва синтаксиса и семантиката на оператор за множествен избор • Реализира програмно условен преход със средствата на конкретен език за програмиране • Конструира ефективни условни конструкции • Проследява изпълнението на разклонен изчислителен процес	• Разклонен изчислителен процес
6.4 *Циклични*	• Описва синтаксиса и семантиката на оператори за цикъл в конкретен език за	

изчислителни процеси	програмиране • Избира подходяща циклична конструкция за решаване на конкретен проблем. • Може да моделира процес описан с една циклична конкструкция чрез друг вид циклична конструкция • Реализира програмно циклични процеси със средствата на конкретен език за програмиране • Проследява изпълнението на цикличен изчислителен процес • Разбира семантиката на вложени циклични конструкции	
7. Капсулиране на данни и методи		
7.1 *Капсулиране на данни и методи*	• Анализира различни практики за моделиране на клас • Следва добри практики за моделиране на клас • Разбира необходимостта от скриване на данни и методи на клас • Описва необходимостта от прилагане на методи за достъп до данните в клас. • Разбира разликата в представянето на променлива от числов и референтен тип данни • Разбира предназначението на конструкторите. • Обяснява предназначението на всеки от конструкторите – по подразбиране, за общо ползване, с параметри и за копиране • Описва конструктори на клас • Използва подходящ конструктор при създаване на обект • Разбира технологията презареждане на методи (*overload*)	• Модификатори за достъп „*public*“ и „*private*“ • Числов тип данни • Референтен тип данни • Интерфейс на обект • Презареждане на метод (*overload*) • Конструктор за общо ползване • Конструктор по подразбиране • Конструктор с параметри

	• Разбира приложението на технологията презареждане на методи при описание и обръщение към конструктори	• Конструктор за копиране
7.2 *Видимост на променливите*	• Описва областта на съществуване на променливите в даден клас • Разпознава локални за даден метод променливи • Описва областта на видимост на променливите в даден клас • Разпознава видимите в дадена област променливи • Генерира диаграма на клас с помощни визуални средства на интегрирана среда • От диаграма на клас генерира изходен код на клас с помощни средства на интегрирана среда	• Област на видимост на променливите • Припокриване на променливи
7.3. *Статични членове на клас*	• Разбира предназначението на статичните членове на клас.Взема обосновано решение за ползване на статичен член на клас • Декларира, инициализира и използва в описанието на класа статични данни • Дефинира статичен метод на клас • Обяснява и прилага техниките за обръщение към статични членове на клас и данни на инстанция. • Разбира необходимостта от използване на константи. • Използва константи на стандартни класове • Познава техники за дефиниране на константи • Дефинира и използва потребителски константи • Дефинира изброим тип данни • Използва данни от изброим тип	• Статичен член на клас • Данни на инстанция • Константа • Изброим тип данни
8. Проектиране на графичен интерфейс		

8.1 *Проектиране на графичен интерфейс*	• Проектира графична форма, представяща интерфейс на конкретно приложение. • Моделира свойства на компоненти от графичния интерфейс етикет, едноредово текстово поле, многоредова текстова област, бутон, компоненти за избор, кутия за изображения, меню, контейнери и др. • Спазва добри практики за именуване на идентификаторите на графични компоненти	• Компонента за избор в графична форма • Многоредова текстова област • Контейнер
9. Класове с потребителски типове данни		
9.1. *Връзки между класове*	• Разбира основни връзки между класове – наследяване, генерализация, асоциация, агрегация, композиция	• Наследяване • Генерализация • Асоциация • Агрегация • Композиция
9.2. *Обектите като членове на класове*	• Проектира клас, в който член-променливите са обекти от друг клас • Декларира член данни на клас, представляващи обекти от друг клас • Използва обекти на един клас в качеството им на член-променливи в други класове • Обяснява технологията на конструиране на обект, в който има данни от друг клас • Различава променливи и непроменливи референтни данни • Дефинира и използва изменяеми и неизменяеми референтни данни на клас	• Релация "*има*" ("*has a*") • Неизменяеми референтни данни на клас • Изменяеми референтни данни на клас

	• Конструира диаграми на класове, намиращи се в релация "*има"* • Реализира примери на релация „*има"* между класове при реализацията на графичния потребителски интерфейс • Познава особеностите на писането на свойствата на изменяеми референтни данни на обекти • Осигурява защита на данните чрез дефиниране на методи за достъп и контролирана промяна на референтни член-данни	
9. 3. *Модулна организация на класове*	• Разбира особеностите на вложени, локални и анонимни класове • Създава библиотека от потребителски дефинирани класове • Разбира предимствата на обединяване на класове в библиотека	• Видимост на класове
10. Наследяване.		
10.1. *Наследяване на класове*	• Разбира същността на релация „***е"*** между два класа • Разбира технологията на наследяване на членове от базов клас при демонстриране на примери • Разбира технологията на наследяване в рамките на йерархия от наследственост • Познава основни методи на клас *Object* – проверка за „равни" обекти, представяне на обект във вид на символен низ • Реализира наследяване на базов клас • Използва наследени методи на базов клас	• Релация „*е"* (*is a*) • Базов клас • Наследник
10. 2. *Капсулиране на данни при наследяване*	• Реализира потребителска йерархия на наследственост • Разбира правилата за онаследяване на член-променливи и методи от пряк и непряк базов клас • Спазва принципа за капсулиране на член-променливи при наследяване	• Йерархия на наследственост • Директен базов клас • Индиректен базов клас • Директен наследник

	• Разбира технологията на предефиниране на методи на базов клас • Предефинира методи на базов клас • Разбира принципа на конструиране на обект от наследен клас • Спазва правилата при описание на конструктори на наследници • Познава средство за достъп до данни и методи на директен базов клас	• Индиректен наследник • Модификатор за достъп *protected* • Предефиниран метод • Референция за достъп до методи и данни в директен базов клас
10. 3. *Полиморфизъм*	• Разбира същността на полиморфизма • Разбира технологията на преобразуване „*нагоре*“ (*upcasting*) на производен клас • Прилага добри практики при преобразуване *„надолу“ (downcasting)* • Реализира полиморфно поведение на референция към базов тип • Познава синтаксиса и семантиката и използва оператор за проверка на съвместимост с типа на преобразуването „*надолу*“	• Полиморфизъм • Преобразуване нагоре (*upcasting*) • Преобразуване надолу
10. 4. *Абстрактен клас*	• Обяснява ролята на абстрактния клас в обектно ориентирания модел • Разбира предназначението на абстрактните методи • Декларира абстрактен метод в базов клас • Реализира абстрактен метод на базов клас в неговите наследници • Използва абстрактен клас за обработка на множество от обекти-инстанции на класове от йерархия на наследственост • Използва свойствата на абстрактен клас при преобразуване „*нагоре*“ и *„надолу“* в рамките на йерархия на наследственост	• Виртуален метод • Абстрактен клас
10. 5. *Интерфейс*	• Обяснява предназначението на чист абстрактен клас	• Интерфейс

	• Обяснява предназначението на интерфейса в обектно ориентирания модел • Осъзнава ролята на интерфейса в обектно ориентирания модел • Посочва стандартни библиотеки за интерфейси • Описва потребителски интерфейс • Придържа се към стандартен стил за именуване на интерфейс • Описва класове, реализиращи интерфейси • Обработва множество от инстанции на класове в йерархия на наследственост посредством интерфейс • Прилага интерфейс с преобразуване на тип „*нагоре*“ (*upcasting*) и „*надолу*“ (*downcasting*) в йерархия на наследственост	• Реализация на интерфейс
11. Обработка на събития		
11.1. *Обработка на събития*	• Посочва основни събития с мишката • Описва обект на събитие на мишката • Асоциира събитие на мишката с функционалност на приложението • Посочва основни събития на клавиатурата • Описва обект на събитие на клавиатурата • Асоциира събитие на клавиатурата с функционалност на приложението	• Събитие • Обект на събитие

Годишен брой часове за изучаване на модула в XI клас - 72 часа през първи учебен срок

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.
- 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**	**40%**
За преговор	**до 6 часа**	
Практически дейности/лабораторни упражнения	**до 36 часа**	**60%**
За контрол и оценка (За входно и изходно ниво, текуща проверка и оценка на знанията)	**до 6 часа**	

СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ

Форми на оценяване:

- **Устна**

Устната форма на оценка е удачна при групови обсъждания, дискусии, генериране на идеи. Оценяват се мнението и аргументите на ученика.

- 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 изчисление на израз, задачи за проследяване на работата на алгоритъм или програмен фрагмент, задачи за откриване на логически грешки в алгоритъм или програмен фрагмент. Писмената форма на изпитване може да се провежда самостоятелно или в съчетание с практическа форма.

- **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 база на портфолио, съдържащо резултатите от практическа работа в клас и защитена домашна работа. Желателно е оценката да отразява всички етапи от практическата разработка (анализ, модел, програмиране, защита и др.), 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%**
Оценки от контролни и работи	**20%**
Оценка на изходно ниво	**20%**

ДЕЙНОСТИ И МЕЖДУПРЕДМЕТНИ ВРЪЗКИ

Дейност №	**Описание**
1.	Създаване на проект на конзолно приложение с вход и форматиран изход на данни. Развиват се дигитални компетентности.

2.	Създаване на приложение с използване на диалогови прозорци за въвеждане и извеждане на данни. Развиват се дигитални компетентности.
3.	Графично приложение за въвеждане и извеждане на данни с графичен интерфейс. Развиват се дигитални компетентности и се осъществява междупредметна връзка с изобразителното изкуство.
4.	Приложение на графичен интерфейс за въвеждане на данни, обработка на събитие натискане на бутон, при което към въведените данни се прилагат аритметични и логически операции и накрая в графичния интерфейс се извежда резултатът. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката и изобразителното изкуство.
5.	Приложение на методите на библиотечен клас Math. Генериране на случайни стойности с библиотечен клас Random. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката.
6.	Прилагане в метод на клас на алгоритъм за намиране на сума, произведение, средно аритметично, най-голям и най-малък елемент, преброяване на повторения в серия от данни, въвеждани от графичен интерфейс. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката и изобразителното изкуство.
7.	Създаване на прост калкулатор за работа с рационални числа. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката.
8.	Създаване на приложение-игра с графичен интерфейс, тип теглене на късметче. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката и изобразителното изкуство.
9.	Моделиране на регистрационна форма. Развиват се дигитални компетентности и се осъществява междупредметна връзка с изобразителното изкуство.
10.	Създаване на приложение на графичен интерфейс за създаване на обекти, чиито данни се въвеждат с текстови полета. Развиват се дигитални компетентности и се осъществява междупредметна връзка с предприемачеството и изобразителното изкуство.
11.	Създаване на графичен потребителски интерфейс чрез наследяване на вградени класове. Развиват се дигитални компетентности.

12.	Създаване на графично приложение, прилагащо полиморфизъм при моделиране на обекти от биологията, физиката, химията, реалния свят. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката, природните науки и изобразителното изкуство.
13.	Графично приложение, моделиращо геометрични фигури чрез прилагане на абстрактни класове и полиморфизъм. Развиват се дигитални компетентности и се осъществява междупредметна връзка с математиката и изобразителното изкуство.
14.	Приложение за рисуване с мишката в графична форма. Развиват се дигитални компетентности и се осъществява междупредметна връзка с изобразителното изкуство.
15.	Приложение за рисуване с клавиатурата в графична форма. Развиват се дигитални компетентности и се осъществява междупредметна връзка с изобразителното изкуство.

Приложение № 82 към т. 82

УЧЕБНА ПРОГРАМА ПО ИНФОРМАТИКА
(ПРОФИЛИРАНА ПОДГОТОВКА)

МОДУЛ 2. „СТРУКТУРИ ОТ ДАННИ И АЛГОРИТМИ“

КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА

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 XI 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. Държавният образователен стандарт (ДОС) - изискванията 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.
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 в последствие 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 XI клас. Разработена е в съответствие с цитираните по-горе система от фактори (1 - 7) и определя:
    - очакваните резултати от обучението в Модул 2 за XI клас в съответствие с ДОС;
    - обема и структурата на учебното съдържание, представени;
    - специфични изисквания за провеждане на обучението;
    - препоръчително разпределение на часовете;
    - форми и методи за оценяване.

УЧЕБНО СЪДЪРЖАНИЕ

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
1. Алгоритми и методи на класове		
1.1. Реализация на алгоритми чрез методи	• Описва методи, реализиращи алгоритъм за специфично поведение на обект от даден клас • Разбира технологията на обръщение към "предефиниран"	• Предефиниране (*override*) на метод

	(*override*) метод • Разбира разликите между “презареждане” (*overload*) и “предефиниране” (*override*) на методи	
1.2. Рекурсия	• Разграничава основни видове рекурсия • Намира рекурентни зависимости • Разбира програмната техника рекурсия • Формулира гранични условия на рекурсия • Описва рекурсивни методи • Описва рекурсивни методи итеративно • Описва итеративни методи рекурсивно	• Рекурсия • Пряка и косвена рекурсия • Гранично условие • Рекурсивен метод
2. Класификация на структурите от данни		
2.1. Абстрактни типове от данни	• Обяснява понятието абстрактен тип данни • Класифицира основните абстрактни типове данни • Познава колекции, реализиращи абстрактни типове данни	• Абстрактен тип данни (АТД) • Колекция
2.2. Класификация на структурите от данни	• Изброява основни структури от данни: масив, символен низ, линеен списък (едносвързан, двусвързан, цикличен), стек, опашка, граф, кореново дърво • Познава структурата, характерни методи и приложения на линейни абстрактни типове данни • Познава структурата, характерни методи и приложения на речници, хеш-таблици и множества • Познава нелинейни (разклонени) АТД	• Линейни АТД • Разклонени АТД • Речник • Хеш таблица • Множество • Граф • Дърво

	• Познава структурата, характерни методи и приложения на граф • Познава структурата, характерни методи и приложения на дърво	
3. Символен низ		
3.1. Символен низ	• Познава стандартен клас, реализиращ символен низ • Декларира, създава и инициализира символен низ	• Символен низ • Празен низ • Дължина на низ
3.2. Обработка на символни низове	• Познава вградени методи за обработка на символен низ • Сравнява низове лексикографски • Използва символни низове за обработка на текст: - изтрива подниз - вмъква подниз - заменя един подниз с друг • Търси подниз в низ по пълно съвпадение: - 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	• Лексикографско сравнение • Конкатенация • Подниз
3.3. Регулярни изрази	• 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	• Метасимвол • Регулярен израз

	• Търси подниз в низ по шаблон • Валидира данни с помощта на регулярни изрази	
4. Едномерен масив		
4.1. Едномерен масив	• 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	Размерна масив
4.2. Сортиране на масив	• 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	• Сортиране на масив • Възходящо/низходящо сортиране
4.3. Техники за работа с	• Прилага алгоритъма за двоично търсене в сортиран масив	• Двоично търсене

сортиран масив	• Обосновано избира 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	• Сливане на сортирани редици
4.4. Комбинаторни алгоритми	• Генерира комбинаторни конфигурации със и без повторение • Прилага комбинаторни алгоритми в игрови ситуации • Моделира и реализира решение на проблем с помощта на комбинаторни алгоритми	• Комбинаторни конфигурации • Вариации • Пермутации • Комбинации
5. Многомерен масив		
5.1. Многомерен масив	• Осмисля структурата на многомерен масив • Изброява възможни приложения на многомерен масив	• Многомерен масив
5.2. Двумерен масив	• 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 • Използва двумерен масив при моделиране на решение на проблем	• Двумерен масив • Ред • Стълб • Диагонал
6. Линейни структури от данни		
6.1. Линейни структури от данни	• Познава общите характеристики на линейни структури от данни • Познава колекции, реализиращи линейни структури от данни	• Списък • Стек

		• Опашка
6.2. Списък	• Разбира структурата от данни вектор • Разбира структурата от данни линеен едносвързан 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	• Итератор
6.3. Стек	• 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	• Връх на стек
6.4. Опашка	• 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	• Начало на опашка • Край на опашка
7. Структури от данни – съпоставка и препоръки		

7.1. Сложност на алгоритъм	• Знае как да оценява сложността на алгоритъм - 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	• Сложност на алгоритъм • Сложност по време • Сложност по памет • Сложност по среден брой операции
7.2. Сравнение на основните структури	• Сравнява структурите по време за: - 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	• Константна сложност • Логаритмична сложност • Линейна сложност • Полиномиална сложност • Експоненциална сложност
7.3. Структури от параметризирани данни	• 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	• Шаблонни класове (generics) • Параметър за тип
8. Файлове и потоци от данни		
8.1. Потоци	• Знае какво представляват потоците и тяхното предназначение	• Поток

	• 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 • Умее да форматира подходящо изходните данни	• Входен поток • Изходен поток • Двоичен поток • Текстов поток
8.2. Методи за работа с потоци	• Обяснява стандартните класове за работа с потоци • Прилага методи за обработка на поток: “създаване”; “отваряне”; “затваряне”	• Създаване • Отваряне • Затваряне
8.3. Текстови файлове	• 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	• Четене • Запис • Добавяне • Търсене
8.4. Сериализация на данни	• 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	• Сериализация • Десериализация
8.5. Приложение	• 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	• 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	• Изключение
9.2. Обработка на грешки	• 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	• Прихващане на изключение • Предизвикване на изключение • Обработка на изключение
9.3. Приложения	• Прилага обработка на изключения при валидиране на потребителски вход. • Прилага регулярни изрази при обработка на изключения при валидиране на данни	• Валидиране на данни
10. Качествен програмен код		
10.1. Именуване на елементите от програмата	• Знае какво е качествен програмен код • Обяснява как да се именуват идентификаторите на: - класове, интерфейси и методи	• Външно качество (през призмата на потребителя) • Вътрешно качество (вътрешна

	- параметри на методи, променливи и константи	организация)
10.2. Правила за форматиране и подреждане на кода	• 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	• Автоматично форматиране на кода • Код-конвенции за форматиране
10.3. Добри практики за изграждане на висококачествени методи	• 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 • Разбира понятията „обхват“, „живот“ и „активност” на променлива • Правилно използва изрази • Прави проверка за коректност на входните данни • Изследва ефективността на методи с еднаква функционалност в различни реализации	• Обхват, живот, активност на променлива • Израз с една операция • Контрол на входните данни
10.4. Принципи за качествена документация на кода	• Създава самодокументиращ се код • Вмъква подходящи коментари	• Самодокументиращ се код • Преработка на код (refactoring)

	• Преработва код	
10.5. Верификация и валидация на цялостно решение	• 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)	• Black box testing • White box testing • Gray box testing

Годишен брой часове за изучаване на модула в XI 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.
- 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**	

СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ

Форми на оценяване:

- **Устна**

Устната форма на оценка е удачна при групови обсъждания, дискусии, генериране на идеи. Оценяват се мнението 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.

- **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 база на портфолио, съдържащо резултатите от практическа 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 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 на 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, 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 *p*-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 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.	Организират 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 програма. 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,... Реализират междупредметни връзки с природонаучните дисциплини - физика, химия, биология, география и история.
Дейност 17.	Представяне и изчертаване на многоъгълник. Намиране лице на многоъгълник. Осъществят 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 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 географията. 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 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.	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. Осъществява се междупредметна връзка с математиката.
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 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 обществено полезни теми.

Приложение № 83 към т. 83

УЧЕБНА ПРОГРАМА ПО ИНФОРМАТИКА
(ПРОФИЛИРАНА ПОДГОТОВКА)

МОДУЛ 3. „РЕЛАЦИОНЕН МОДЕЛ НА БАЗИ ОТ ДАННИ“

КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА

Обучението в **Модул 3. „Релационен модел на бази от данни”** е насочено към овладяване на базисни знания, умения и отношения, необходими за софтуерната реализация на информационни системи с бази от данни в съвременното общество и с изграждането на основни познавателни, приложни и аналитични компетентности на ученика в тази област. Учебната програма е предвидено да се реализира в рамките на 72 учебни часа за ученици в XII клас през първия учебен срок. Учебната програма е представена в следната последователност:

1. Учебно съдържание и очаквани резултати от обучението
2. Хорариум на модула
3. Оценяване
4. Дейности и междупредметни връзки

Целта на учебната програма е да направи съвременна интерпретация на типични информационни проблеми, посредством достъпно за учениците съчетание на помощни визуални средства с приложение на ефективни технологии за обектно ориентирано програмиране. Учебното съдържанието е обособено в 2 основни теми (“Информационни системи с бази от данни”, “Моделиране и обработване на данни“) всяка, от които е структурирана в подтеми, проектиращи елементи от стандартите в ДОИ в система от понятия и взаимовръзките им с методите за приложение на софтуерни технологии за реализиране на очакваните резултати. Софтуерните технологии са подбрани като са взети предвид следните взаимосвързани фактори:

- Знанията, уменията и отношенията, изградени в предходните модули на профилираната подготовка по информатика

- Възможностите да се постави акцент върху логически връзки и моделиране с използване на утвърдени стандарти в софтуерните технологии, а писането на програмен код да се минимизира
- Възможността за постигане на ефективна и изпълнима от ученици програмна реализация на съвременни информационни проблеми, посредством широкото приложение на стандартни софтуерни библиотеки в кратка последователност от технологични стъпки
- Стандартното софтуерно и хардуерно осигуряване за профилирана подготовка в системата на средното образование – компютри, системен и приложен софтуер, мрежова и комуникационна инфраструктура
- Възможностите за достъпно представяне на тези технологии пред ученици с мултимедийни и други съвременни дидактически средства
- Съвременните тенденции в образователната политика и връзката им с изискванията на обществото

Очакваните резултати от приложението на учебната програма са всеки ученик да придобие основни компетентности за разработка на информационни системи с бази от данни и овладяване на базисни знания за продължаване на обучението в областта на компютърните науки. Постигането на тези резултати изисква въвличането на учениците в ролята им на активни участници в учебния процес с помощта на съвременни средства за обучение и съвременни педагогически практики.

УЧЕБНО СЪДЪРЖАНИЕ

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
1. Информационни системи с бази от данни		
1.1 Въведение в информационните системи.	• Разграничава характерни функционалности за основни видове Информационни системи • Коментира предимства и недостатъци на информационни системи, базирани на файлове и електронни таблици • Знае характерните особености на информационни системи с бази от данни	• Информационна система • Система за управление на човешките ресурси (HRMS) • Система за обработка на транзакции (TPS)

	Назовава приложението на таблици, заявки и отчети в информационни системи с бази от данни • Оперира с примерна информационна система с база от данни	• Система за управление на информацията (MIS) • Дублиране на данни (data redundancy) • Споделяне на данни • Сигурност на база от данни • Заявка • Отчет • Едновременен достъп на потребители до бази от данни
1.2 Системи за управление на бази от данни (СУБД)	• Описва предназначението и ролята на компонентите на СУБД • Разпознава архитектури за реализиране на многопотребителски СУБД. • Познава и разбира три нива на абстракция на СУБД: външно, концептуално и вътрешно ниво • Описва основните роли на групите потребители на СУБД	• Система за управление на база от данни (СУБД) • Сървър на база от данни, • Архитектура на база от данни • Клиент - сървър архитектура • Файлов сървър • Разпределена система • Външно ниво на абстракция • Концептуално ниво на

		абстракция • Вътрешно ниво на абстракция • Клиентско приложение на бази от данни • Схема на бази от данни • Процедура на СУБД • Доставчик на база от данни • Източник на данни
1.3 Релационен модел на база от данни	• Познава структурата и характеристиките на релационния модел на база от данни • Познава основните термини и понятия, свързани с релационен модел на база от данни • Използва основни термини и понятия за представяне на примери на релационни бази от данни • Структурира и представя примерно описани данни чрез релационен модел	• Релационен модел на база от данни • Релация/отношение (таблица) • Атрибут (поле, колона) • Домейн на атрибут (множество от стойности) • Степен на отношение (брой атрибути) • Кортеж (запис, ред) • Мощност (брой кортежи)
1.4 Представяне на данните	• Създава таблица в интегрирана среда (MS Access) с графични средства като прилага добри практики за именуване на таблица и полета на таблица.	• Целочислен тип данни • Тип данни с плаваща

	• Разграничава приложението на основни типове данни на полета на таблица. • Прилага подразбиращи се стойности и проверка за област на допустими стойности • Въвежда и редактира данни в таблица с графични средства • Знае за възможностите за преобразуване от един тип в друг • Задава стойност NULL на данни	запетая • Тип данни символен низ • Тип данни за дата и час • Стойност NULL
2. Моделиране и обработване на данни		
2.1 Видове релации	• Обяснява основните видове степени на отношения (relationship) • Познава видовете връзки/отношения • Свързва основен и външен ключ при описание на релация • Задава основен ключов атрибут на релация • Задава външен ключов атрибут на релация • Създава прости и съставни основни и външни ключове на релация • Разбира задаването на типове ограничения (not NULL, Unique, Primary key, Foreign key и т.н.) за даден атрибут • Интерпретира основни графични означения в диаграма на бази от данни • Разпознава основните степени на релация и видове връзки в ER диаграма • Илюстрира основните степени на релации и видове връзки при решаване на информационен проблеми	• Унарни релации • Бинарни релации • Релации от по-висока степен • Връзки/отношения от вида 1:1, 1:M, N:M, - • Основен ключ (primary key) • Автоматично генериране на стойност на първичен ключ • Външен ключ • Прост ключ • Съставен ключ

		• Графичен модел на бази от данни - Entity Relationship (ER) диаграма
2.2 Моделиране на бази от данни	• Разпознава аномалии от повторение на група от данни в таблица • Прилага основни правила за избягване на аномалии • Открива аномалии, породени от излишество на данни в графичен модел на база от данни и предлага начини за отстраняването им • Разбира процеса на нормализация на релационна база от данни • Прилага нормализация в първа и втора нормална форма (1NF, 2NF) на релационна база от данни • Моделира диаграма на релационна база от данни с графични средства по примерно описание на информационна структура от категории данни	• Аномалии при вмъкване, редактиране и изтриване на записи, породени от излишество на данни • Транзитивно изчислимо поле • Нормална форма на база от данни • Първа нормална форма на база от данни • Втора нормална форма на база от данни • Логическо и физическо представяне на база от данни
2.3 Основни операции в релационен модел	• Познава основните операции в релационен модел • Идентифицира основни операции в релационен модел на дадена база от данни • Прилага конкретна основна операция над дадена база от данни	• Проекция • Селекция • Декартово произведение

	• Идентифицира основни операции и реда на тяхното приложени при решаване на практически задачи	• Обединение • Сечение • Разлика • Деление • Групиране • Агрегатни функции
2.4 Моделиране и изпълнение на заявки	• Описва характерни особености и приложения на заявките в СУБД • Моделира и изпълнява с графични средства заявка за извличане, реализираща проекция или селекция върху една таблица • Представя с помощта на графични средства резултатът от заявка в нарастващ или намаляващ ред на стойностите на дадено поле • Позволява или забранява наличието на дублирани редове в заявка • Задава с помощта на графични средства критерии за селекция и съединение • Илюстрира с графични средства заявки с решения на несложни информационни проблеми	• Заявка • Критерий на заявка
2.5 Заявки със свързани таблици от данни	• Познава и използва основни операции за обновяване на базата от данни – добавяне на редове, изтриване на редове, модифициране на стойности • Прилага основните видове интегритет на свързване (referential integrity) Моделира и изпълнява с графични средства заявка за извличане, реализираща проекция, селекция и съединение върху две или повече таблици. • Моделира и изпълнява с графични средства вложени заявки	• Естествено свързване (natural join) на две или повече таблици • Вътрешно свързване (inner join) на две или повече таблици • Интегритет на свързване при изтриване и

		редактиране
2.6 Стандартен език за работа с релационни бази от данни	• Описва приложенията на стандартен език за работа с релационни бази от данни (SQL) • Идентифицира ключови думи от команди на SQL за вмъкване, извличане, редактиране и изтриване на информация • Разчита SQL заявка, генерирана с графични средства • Генерира заявки за извличане, редактиране и изтриване на записи в таблица чрез графични средства. • Подбира подходящи дейности (Create, Read, Update and Delete – CRUD) за управлението на примерна база от данни (вмъкване, прочитане, редактиране и изтриване на записи в таблици) при решаване на зададен проблем	• Стандарти за представяне на данни и заявки в SQL
2.7 Заявки с потребителски дефинирани изчисления	• Дефинира изчислимо поле на заявка • Прилага стандартни аритметични и текстови операции с полета на таблица • Използва изчислимо поле на заявка, създадена с графични средства или със средствата на SQL, за решение на примерни задачи в СУБД • Описва вложена заявка, включваща изчислимо поле •	• Изчислимо поле на заявка • Заглавие на изчислимо поле
2.8 Заявки за получаване на агрегирани данни	• Идентифицира приложения на заявки за извличане на агрегирани данни в СУБД • Подбира математически функции за получаване на агрегирани данни • Използва групиране на записи за извличане на агрегирани данни • Описва с помощта на графични средства заявки за извличане на агрегирани	• Агрегирани данни • Групиране (GROUP BY) • Условие за група (HAVING) • Подредба на редовете от

	данни • Описва с помощта на SQL заявки за извличане на агрегирани данни	резултата (ORDER BY)
2.9 Заявки с параметри	• Идентифицира приложения на параметричните заявки в СУБД • Генерира със средствата на Access заявки с параметри при извличане, редактиране и изтриване на записи като използва помощни графични средства • Анализира записи от данни, чрез промяна на стойностите на определени параметри на заявка	• Параметър на заявка • Параметрична заявка

Годишен брой часове за изучаване на модула в XII 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.
- 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 описаните нови понятия да се разглеждат на концептуално ниво и да се използват при практическата работа.
- Часовете за преговор да включват предимно изпълнение на компютър на практически дейности по теми от учебната програма.

- Препоръчително е да се използват примерни бази от данни на MS Access и MySQL и Microsoft SQL Server с интегрираните среди NetBeans и Visual Studio .NET.
- Да се идентифицират и прилагат добри практики на програмиране
- 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 30 часа**	**до 50%**
За преговор	**до 15 часа**	
Практически дейности	**до 18 часа**	**до 50%**
За контрол и оценка (За входно и изходно ниво, междинно контролно)	**до 6 часа**	

СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ

Съотношение при формиране на срочна и годишна оценка:

Текущи оценки от практически задания в клас	**20%**
Текущи оценки от домашни работи	**20 %**
Оценки от работа по проект	**10%**
Оценка на изходно ниво	**30%**
Оценки от контролни работи	**20%**

ДЕЙНОСТИ И МЕЖДУПРЕДМЕТНИ ВРЪЗКИ

Дейности	Описание на дейност
Дейност 1.	Структуриране на примерно описани масиви от данни в таблици на релационна база от данни. Учениците прилагат добри практики за създаване на таблица - именуват полетата на таблиците, демонстрират познавателни, аналитични и приложни компетентности при определяне и задаване на тип на поле на таблица, стойност по подразбиране, ограничения за валидност на данните, както и задаване на основен ключ на таблица. Удачни примери за реализация на тази дейност могат да се открият в теми от предмета по предприемачество, както и в други предмети от училищната програма, където има нужда от обработка на динамично генерирани и свързани помежду си данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект
Дейност 2.	Моделиране на основните видове релации (1:1, 1:M) между таблици като се демонстрирта познавателни и приложни компетентности при прилагане на основни правила за избягване на аномалии при вмъкване, редактиране и изтриване на записи . Учениците демонстрират аналитични и приложни компетентности при моделиране на диаграма на релационна база от данни с графични средства по примерно описание на информационна структура от категории данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект
Дейност 3.	Въвеждане и редактиране на данни в таблица с графични средства. Учениците демонстрират познавателни и приложни компетентности за валидиране на данни, използване на подразбиращи се стойности и въвеждане на външен ключ, чрез „подсказване”(lookup) на набор от възможни стойности с графични средства по примерно описание на информационна структура от категории данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект
Дейност 4.	С графични помощни средства моделиране и изпълняване на заявка за извличане, сортиране и филтриране на записи от таблица. Учениците демонстрират познавателни и приложни компетентности за създаване на заявка, която интерпретира предварително формулирана цел за обработка на информация. Реализацията на тази дейност може да използва данни и задания за заявки, свързани с изследвания върху процеси, факти и събития в областите на природоматематическите и хуманитарните предмети. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за

	формулиране на проект
Дейност 5.	Модифициране с графични средства на заявка върху свързани таблици от данни при анализ на информационни проблеми от вида „Какво, ако?” Учениците демонстрират аналитични и приложни компетентности по реализацията на основните видове интегритет на свързване (referential integrity) и интерпретират решението на зададен информационен проблем за анализ от вида „Какво, ако?” (What, if?”) с промяната на параметри в проектирането, пресмятането, сортирането, филтрирането или групирането на предварително зададена заявка. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект
Дейност 6.	Генериране на структурирани команди на заявки за извличане, редактиране и изтриване на записи с графични средства. Учениците демонстрират познавателни и приложни компетентности да генерират структурирани заявки на стандартен език (SQL) с помощни средства и да интерпретират логиката, представена чрез структуриране на заявката като я съпоставят по аналогия с използваните графични средства.

Приложение № 84 към т. 84

УЧЕБНА ПРОГРАМА ПО ИНФОРМАТИКА
(ПРОФИЛИРАНА ПОДГОТОВКА)

МОДУЛ 4. „ПРОГРАМИРАНЕ НА ИНФОРМАЦИОННИ СИСТЕМИ“

КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА

Обучението в **Модул 4. „Програмиране на информационни системи”** е насочено към овладяване на базисни знания, умения и отношения, необходими за софтуерната реализация на информационни системи с бази от данни в съвременното общество и с изграждането на основни познавателни, приложни и аналитични компетентности на ученика в тази област. Учебната програма е предвидено да се реализира в рамките на 52 учебни часа за ученици в XII клас през втория учебен срок. Учебната програма е представена в следната последователност:

1. Учебно съдържание и очаквани резултати от обучението
2. Хорариум на модула
3. Оценяване
4. Дейности и междупредметни връзки

Целта на учебната програма е да направи съвременна интерпретация на типични информационни проблеми, посредством достъпно за учениците съчетание на помощни визуални средства с приложение на ефективни технологии за обектно ориентирано програмиране. Учебното съдържанието е обособено в 2 основни теми“Разработване на Информационни системи ”, “Анализ, проектиране и реализиране на примерни приложения ”) всяка, от които е структурирана в подтеми, проектиращи елементи от стандартите в ДОИ в система от понятия и взаимовръзките им с методите за приложение на софтуерни технологии за реализиране на очакваните резултати. Софтуерните технологии са подбрани като са взети предвид следните взаимосвързани фактори:

- Знанията, уменията и отношенията, изградени в предходните модули на профилираната подготовка по информатика

- Възможностите да се постави акцент върху логически връзки и моделиране с използване на утвърдени стандарти в софтуерните технологии, а писането на програмен код да се минимизира
- Наличните визуални помощни средства в съвременните интегрирани среди за обектно ориентирано програмиране на C#.NET и Java
- Възможността за постигане на ефективна и изпълнима от ученици програмна реализация на съвременни информационни проблеми, посредством широкото приложение на стандартни софтуерни библиотеки в кратка последователност от технологични стъпки
- Стандартното софтуерно и хардуерно осигуряване за профилирана подготовка в системата на средното образование– компютри, системен и приложен софтуер, мрежова и комуникационна инфраструктура
- Възможностите за достъпно представяне на тези технологии пред ученици с мултимедийни и други съвременни дидактически средства
- Съвременните тенденции в образователната политика и връзката им с изискванията на обществото

Очакваните резултати от приложението на учебната програма са всеки ученик да придобие основни компетентности за разработка на информационни системи с бази от данни и овладяване на базисни знания за продължаване на обучението в областта на компютърните науки. Постигането на тези резултати изисква въвличането на учениците в ролята им на активни участници в учебния процес с помощта на съвременни средства за обучение и съвременни педагогически практики. Предложената структура от подтеми и хорариум, дейности и междупредметни връзки позволяват периодично след всеки 2-4 учебни часа учениците да усвоят умения и добри практики за практическа реализация на цялостно софтуерно решение на практически проблеми. В края на учебната програма са включени теми, които позволяват на учениците да проявят творчество и самостоятелност при разработка на софтуерни проекти. Тези проекти имат за цел да илюстрират интердисциплинарния характер на информатиката в съвременното динамично и глобално свързано общество.

УЧЕБНО СЪДЪРЖАНИЕ

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
1. Разработване на Информационни системи		
1.1 Конфигуриране на работна среда за	• Прилага стандартни средства за стартиране на сървър на база от данни	• Сървър на база от данни, Клиентско приложение на БД,

програмиране на приложения за СУБД	• Подбира доставчици на бази от данни • Конфигурира източник на данни и създава връзка към релационна база от данни с графични средства на интегрирана среда за програмиране • Описва основните роли на групите потребители на СУБД	• Доставчик на база от данни, • Текстово описание на връзката(connection string) • Източник на данни
1.2 Транзакции на операции с БД	• Обяснява характерни особености на изпълнение на транзакции при паралелен достъп до данни в СУБД • Създава примерна база от данни и таблици с графични средства на интегрирана среда за програмиране • Изпълнява транзакции при въвеждане и редактиране на данни в таблица с графични средства на интегрирана среда за програмиране • Архивира и възстановява база от данни, посредством стандартен език за описване на транзакции • Разбира необходимостта от защита и сигурност на данните	• Паралелен достъп до данни • Транзакция, обхват на транзакция, свойства (ACID) на транзакция, записване на промени от транзакция (COMMIT) и възстановяване на промени от транзакция (ROLLBACK)
1.3 Обектно ориентиран модел на релационна БД	• Описва характерните особености и компонентите на обектно ориентиран модел на база от данни • Назовава предимства и недостатъци в приложението на обектно ориентиран модел на база от данни • Генерира обектно ориентиран модел на зададена база от данни с графични средства на средства на	• обектно ориентиран модел на база от данни • Моделиране на основен ключ • Свойства на данни (get/set) • Клас за представяне на категория (entity) от данни • Клас за управление на категории (entity) от

	интегрирана среда за програмиране • Идентифицира характерни елементи в структурата на генерираните класове от обектно ориентиран модел,	данни
1.4 Моделиране на графичен интерфейс за ИС	• Изготвя програмна реализация на графичен потребителски интерфейс с помощни графични средства. • Моделира графични компоненти (етикет, текстово поле, текстова област, списък, падащ списък, полета за избор на опция, бутони) и разположението им в графичен прозорец като прилага добри практики за програмиране • Програмира събития, породени от натискане на бутон, избор на елемент от списък и на поле с опция • Създава софтуерно приложение за обработка на данни, въведени с графичен потребителски интерфейс и извеждане на получените резултати	
1.5 Графично приложение на обектно ориентиран модел на БД	• Генерира софтуерно приложение за извеждане на записите на таблица от база от данни в таблица на графичен прозорец като се използват графични помощни средства • Свързва графични компоненти (текстово поле, текстова област, списък, падащ списък) към полета на категория от данни като използва графични помощни	• Графична компонента за таблично извеждане на данни (DataGridView / JTable) • Свързване на графична компонента към категория от данни • Синхронизиране на графични компоненти с данни от база от данни

	средства • Синхронизира избора на ред от графичната таблица със стойностите на полетата, извеждани в други графични компоненти с графични помощни средства • Разработва програмна реализация на събитие за запазване на редактираните стойности в полетата от таблицата	
1.6 Графичен интерфейс от вида „*едно-към-много*”	• Генерира обектно ориентиран модел по таблици на зададена база от данни, свързани с релация „едно-към-много”, за моделиране на решение на примерни информационни проблеми • Генерира приложение с графичен интерфейс от вида „едно-към-много” (Master-Detail/Parent-Child) на свързани таблици с релация „едно-към-много” посредством графични помощни средства и обяснява генерирания изходен код • Дискутира приложения, където е подходящо да се използва графичен интерфейс от вида „едно-към-много” • Подбира подходящи компоненти на графичния интерфейс	• Релация „едно-към-много” между класове (entity) на обектно ориентиран модел • Насоченост на релация между класове (entity) на обектно ориентиран модел • Графичен интерфейс от вида „едно-към-много ” (Master-Detail/ Parent-Child)
1.7 Въведение в интегриран език за	• Интерпретира ключови думи на интегриран език за дефиниране на заявки при създаване на структура от	• Интегриран език за дефиниране на заявки, структура от данни на заявка

дефиниране на заявки	данни, съдържаща записи в таблица на източник на данни • Прилага филтриране и сортиране на елементите на заявка посредством интегриран език за дефиниране на заявки • Коментира аналогиите на интегриран език за дефиниране на заявки със стандартния език за изпълнение на структурни заявки в релационна база от данни • Създава приложение на обектно ориентиран модел на база от данни за извеждане записи от таблица с филтриране и сортиране	• Параметър за тип на структура от данни на заявка • Дефиниране на заявка на интегриран език, филтриране и сортиране • Изпълнение на заявка на интегриран език с програмни средства, обект от клас за управление на категории от данни
1.8 Съхраняване на категория (entity) от данни	• Генерира обектно ориентиран модел по таблици на зададена база от данни за моделиране на решение на конкретен проблем • Създава графичен интерфейс за въвеждане на категория от данни като използва графични помощни средства. • Свързва графичните компоненти към съответните полета на категория от данни като използва графични помощни средства • Използва класа за управление на категориите от данни за съхраняване на въведена категория (entity) от данни	• Съхраняване на категория (entity) от данни • Метод за съхранение на категория данни

1.9 Редактиране на категория (entity) от данни	• Генерира обектно ориентиран модел по таблици на зададена база от данни за моделиране на решение на конкретен проблем • Създава графичен интерфейс за редактиране на категория от данни като използва графични помощни средства. • Свързва графичните компоненти към съответните полета на категория от данни като използва графични помощни средства • Използва класа за управление на категориите от данни за съхраняване на редактирана категория (entity) от данни	• Редактиране на категория (entity) от данни • Метод за намиране на категория (entity) от данни за редактиране, потвърждение за редактиране
1.10 Изтриване на категория (entity)	• Генерира обектно ориентиран модел по таблици на зададена база от данни за моделиране на решение на конкретен проблем • Създава графичен интерфейс за изтриване на категория от данни като използва графични помощни средства. • Свързва графичните компоненти към съответните полета на категория от данни като използва графични помощни средства • Използва класа за управление на категориите от данни за изтриване на избрана категория (entity) от данни	• Изтриване на категория (entity) от данни • Метод за намиране на категория (entity) от данни за изтриване, потвърждение за изтриване

1.11 Филтриране и сортиране на данни от свързани категории (entities)	• Генерира обектно ориентиран модел по таблици на зададена база от данни, свързани с релация „едно-към-много”, за моделиране на решение на конкретен проблем • Създава графичен интерфейс към свързани категории от данни като използва графични помощни средства • Подбира и настройва свойствата на графичните компоненти към съответните категории данни • Прилага методи от класа за управление на категориите и интегриран език за дефиниране на заявки за филтриране и сортиране на данни от свързани категории (entities) при програмиране на събитията в графичния интерфейс	
1.12 Обработка на данни от свързани категории (entities)	• Генерира обектно ориентиран модел по таблици на зададена база от данни, свързани с релация „едно-към-много”, за моделиране на решение на примерни информационни проблеми • Създава графичен интерфейс към свързани категории от данни и резултати от обработка на данни като използва графични помощни средства. • Подбира и настройва свойствата на графичните компоненти към съответните категории данни • Прилага методи от класа за управление на категориите	

	и интегриран език за дефиниране на заявки за обработка на данни (пресмятания) от свързани категории (entities) при програмиране на събитията в графичния интерфейс	
2. Анализ, проектиране и реализиране на примерни приложения		
2.1 Анализ на изискванията за информационна система	• Описва етапите в жизнения цикъл на информационна система • Анализира съответствието на изискванията за организация на информацията в зададен пример по отношение на зададен модел на релационни бази от данни и коригира допуснати пропуски в зададения модел • Анализира изискванията за реализация на информационни процеси в конкретен пример • Обобщава, документира и представя изискванията	• Жизнен цикъл на информационна система
2.2 Проектиране и програмиране на информационна система	• Работи в екип при моделиране на решение на част от анализирания проблем, създава обектно ориентиран модел на база от данни, проектира графичен потребителски интерфейс • Интерпретира модел за решаване на примерно зададен информационен проблем	• Архитектура на софтуерно приложение • Добър стил на графичен интерфейс на информационна система

	• Генерира програмен код за реализация на зададен модел с помощта на стандартни библиотеки и графични помощни средства.	
2.3 Тестване и представяне на проект	• Описва работата от етапите на реализацията на софтуерен проект. • Създава тестови примери с входни данни и коментира получените резултати. • Представя проект пред аудитория.	Тестване, автоматизирани средства за тестване, симулация на процес

Годишен брой часове за изучаване на модула в XII клас - 52 часа през втория учебен срок

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.
- 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 описаните нови понятия да се разглеждат на концептуално ниво и да се използват при практическата работа.
- Часовете за преговор да включват предимно изпълнение на компютър на практически дейности по теми от учебната програма.
- Препоръчително е да се използват интегрираните среди NetBeans и Visual Studio .NET.

- За създаване на обектно ориентиран модел на бази от данни се препоръчва да се използва Java Persistence API (JPA) с Java Persistence Query Language (JPQL), EclipseLink Query Language (EQL) или Entity Framework (EF) с Language INtegrated Query (LINQ) посредством съответните помощни графични средства в интегрираните среди за програмиране.
- Да се идентифицират и прилагат добри практики на програмиране
- 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 30 часа**	**до 50%**
За преговор	**до 15 часа**	
Практически дейности	**до 18 часа**	**до 50%**
За контрол и оценка (За входно и изходно ниво, междинно контролно)	**до 6 часа**	

СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ

Съотношение при формиране на срочна и годишна оценка:

Текущи оценки от практически задания в клас	**20%**
Текущи оценки от домашни работи	**20%**
Оценки от работа по проект	**10%**
Оценка на изходно ниво	**30%**
Оценки от контролни работи	**20%**

ДЕЙНОСТИ И МЕЖДУПРЕДМЕТНИ ВРЪЗКИ

Дейности	Описание на дейност
Дейност 1.	Конфигуриране на източник на данни и създаване на връзка към релационна база от данни с графични средства на интегрирана среда за програмиране. Учениците демонстрират познавателни и приложни компетентности при избор на доставчик на база от данни и параметрите, необходими за създаване на връзка към база от данни, както и при използване на помощни средства за стартиране, разглеждане на структурата на базата от данни и спирането на базата от данни.
Дейност 2.	Създаване на база от данни, таблица и въвеждане/редактиране на данни върху сървър от база от данни с помощни визуални средства на интегрирана среда за програмиране. Учениците демонстрират познавателни и приложни компетентности при дефиниране на свойствата на полета на таблица по предварително зададен модел на таблица като съобразяват типовете данни и останалите свойства на полетата на таблицата с изискванията на съответната релационна база от данни, разграничават операции COMMIT и ROLLBACK с транзакции съответно при потвърждаване и отказване на въвеждани данни.
Дейност 3.	Архивиране и възстановяване на зададена база от данни, посредством скрипт на стандартен език (SQL) за структуриране на заявки. Учениците демонстрират умения за използване на помощни средства на интегрирана среда програмиране при архивиране и възстановяване на релационна база от данни върху сървър за база от данни.
Дейност 4.	Генериране на обектно ориентиран модел на зададена база от данни с графични средства на средства на интегрирана среда за програмиране. Учениците демонстрират познавателни и приложни компетентности за структурата на обектно ориентиран модел и начина за генерирането му със средствата на интегрирана среда за програмиране. В комбинация с други от тук споменатите дейности тази дейност може да се използва за формулиране на проект.
Дейност 5.	Генериране на софтуерно приложение за извеждане на записите на таблица от база от данни в таблица на графичен прозорец като се използват визуални помощни средства. Учениците демонстрират познавателни и приложни компетентности по структурата на обектно ориентиран модел на бази от данни и приложението на визуални помощни средства за синхронизиране на таблична компонента от графичния интерфейс със записите на таблица от примерно зададена база от

	данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект
Дейност 6.	Генериране на софтуерно приложение с графичен интерфейс от вида „едно-към-много” (Master-Detail/Parent-Child) на свързани таблици с релация „едно-към-много” посредством графични помощни средства в интегрирана среда за програмиране. Учениците демонстрират познавателни и приложни компетентности как се интерпретира релация „едно-към-много ” в обектно ориентиран модел на така свързани таблици, предимствата на графичен интерфейс от вида „едно-към-много”, както и реализацията на този вид интерфейс с помощните визуални средствата на интегрирана среда за програмиране. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект.
Дейност 7.	Създаване на конзолно приложение на обектно ориентиран модел на бази от данни за извеждане записи от таблица с филтриране и сортиране. Учениците демонстрират познавателни и приложни компетентности при интерпретация на зададени команди на интегриран език за дефиниране на заявки по аналогия с командите на стандартен език за структуриране на заявки (SQL), както и при използване на средствата на обектно ориентиран език за изпълнение на тези команди - примерно команди за извеждане записи от таблица с филтриране и сортиране. В комбинация с други от тук споменатите дейности тази дейност може да се използва за формулиране на проект.
Дейност 8.	Създаване на интерактивен графичен интерфейс за въвеждане на категория от данни като използва графични помощни средства на интегрирана среда. Учениците демонстрират познавателни и приложни компетентности при подбор на графичните компоненти и помощните средства за синхронизацията им с базата от данни, а също и средствата на обектно ориентиран език за съхранение на категория от данни в съответната й таблица на базата от данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект.
Дейност 9.	Създаване на интерактивен графичен интерфейс за редактиране на категория от данни като използва графични помощни средства на интегрирана среда. Учениците демонстрират познавателни и приложни компетентности при подбор на графичните компоненти и помощните средства за синхронизацията им с базата от данни, а също и средствата на обектно ориентиран език за редактиране на категория от данни и последващото й съхранение в съответната й таблица на базата от

	данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект.
Дейност 10.	Създаване на интерактивен графичен интерфейс за изтриване на категория от данни като използва графични помощни средства на интегрирана среда. Учениците демонстрират познавателни и приложни компетентности при подбор на графичните компоненти и помощните средства за синхронизацията им с базата от данни, а също и средствата на обектно ориентиран език за изтриване на категория от данни от съответната ѝ таблица на базата от данни. В комбинация с други от тук споменатите дейности, тази дейност може да се използва за формулиране на проект.
Дейност 11.	Създаване на графичен интерфейс към свързани категории от обектно ориентиран модел на данни като използва визуални помощни средства. Учениците демонстрират познавателни и приложни компетентности да използват помощни средства на интегрирана среда, за да генерират релация „едно-към-много ” в обектно ориентиран модел на база от данни при решение на примерни информационни проблеми, както и да прилагат методи от класа за управление на категориите в обектно ориентиран модел на базата от данни и интегриран език за дефиниране на заявки за извличане и обработка на информация (филтриране, сортиране и изчисления) от свързани категории (entities) при програмиране на събитията в графичния интерфейс.
Дейност 12.	Създаване, документиране, тестване и представяне на проект по програмиране на информационни системи. Учениците демонстрират аналитични, познавателни и приложни компетентности при документиране на изискванията за изпълнение на проекта и етапите в жизнения му цикъл, генериране на програмен код за реализация на зададен модел с помощта на стандартни библиотеки и визуални помощни средства, тестване на полученото приложение, както и при структуриране и докладване на извършената работа.