



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 1**

---

\* 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 1 “ОБЕКТНО-ОРИЕНТИРАНО ПРОЕКТИРАНЕ И ПРОГРАМИРАНЕ”

#### КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА НА МОДУЛА

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 11. клас е съобразена със система от взаимосвързани 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 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.



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

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 профилираща подготовка във втори гимназиален етап. Съдържанието е предвидено да се реализира в рамките на 72 учебни часа в първи учебен срок на 11. клас. Предназначена е за учещи в математическите гимназии на Република България. Разработена е в съответствие с цитираната по-горе система от фактори (1 - 7) и определя:
  - очакваните резултати от обучението в модул 1 за 11. клас в съответствие с ДООИ;
  - 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
<b>1. Въведение в информатиката</b>		
<b>1.1 Езици и среди за програмиране</b>	<ul style="list-style-type: none"> <li>• Описва основните групи езици за програмиране и обяснява предназначението им</li> <li>• Разбира предназначението на интегрирана среда за програмиране</li> <li>• Разпознава видове езици за програмиране според тяхното ниво</li> <li>• Разграничава независим от зависим от средата и/или платформата програмен код</li> </ul>	<ul style="list-style-type: none"> <li>• Стандартен вход</li> <li>• Стандартен изход</li> <li>• Типизирани и нетипизирани езици за програмиране</li> <li>• Език за структурно програмиране</li> <li>• Език за обектно-ориентирано програмиране</li> <li>• Език за функционално</li> </ul>



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

		<p>програмиране</p> <ul style="list-style-type: none"> <li>• Език за логическо програмиране</li> <li>• Събитийно-ориентирано програмиране</li> <li>• Машинен език</li> <li>• Асемблерни езици</li> <li>• Езици от високо ниво</li> <li>• Транслатор</li> <li>• Компилятор</li> <li>• Интерпретатор</li> </ul>
<b>1.2. Обектно-ориентиран подход</b>	<ul style="list-style-type: none"> <li>• Дефинира понятието клас от гледна точка на обектно-ориентирания подход</li> <li>• Дава примери на ситуации от реалния свят с използване на обектно ориентирания подход.</li> <li>• Разбира различните нива на абстракция при проектиране на клас – взаимозависимост и свързаност, достатъчност, пълнота и простота</li> <li>• Обяснява връзката между класове чрез примери</li> <li>• Разграничава състояние и поведение на обектите на клас</li> </ul>	<ul style="list-style-type: none"> <li>• Обектно-ориентиран подход</li> <li>• Ниво на абстракция</li> </ul>
<b>2. Класове и обекти</b>		
<b>2.1. Интегрирана среда за обектно-ориентирано програмиране (ООП)</b>	<ul style="list-style-type: none"> <li>• Посочва основните компоненти на интегрирана среда за програмиране</li> <li>• Посочва компонентите на проект в интегрирана среда за програмиране</li> <li>• Използва интегрирана среда за програмиране за разработка на софтуерен (програмен) проект</li> <li>• Познава процеса на преобразуване на изходен код на програмен проект до изпълнима програма</li> </ul>	<ul style="list-style-type: none"> <li>• Проект</li> <li>• Специализиран текстов редактор</li> <li>• Свързващ редактор</li> <li>• Редактор за откриване и отстраняване на грешки</li> <li>• Инструмент за преобразуване до изпълним код</li> </ul>



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"> <li>• Редактор за проектиране на дизайн на графичен потребителски интерфейс</li> </ul>
<b>2.2. Основни елементи на език за обектно ориентирано програмиране</b>	<ul style="list-style-type: none"> <li>• Различава ключови, стандартни и потребителски думи</li> <li>• Изброява ключови думи и основни елементи на обектно-ориентиран език за програмиране</li> <li>• Използва конвенция за именуване на клас, метод и променливи (стил „камила“ (CamelCasing) и „Паскал“ (PascalCasing))</li> <li>• Дефинира понятията клас, обект, член-променливи на клас, член-функции (методи) на клас в обектно-ориентиран език за програмиране</li> <li>• Различава настолни приложения с графичен и конзолен интерфейс, уеб приложения, приложения за мобилни устройства и вградени системи</li> </ul>	<ul style="list-style-type: none"> <li>• Идентификатор</li> <li>• Ключова дума</li> <li>• Стандартна дума</li> <li>• Потребителска дума</li> <li>• Конвенция за имената</li> <li>• Клас</li> <li>• Член-променлива</li> <li>• Член-функция (метод)</li> <li>• Обект</li> <li>• Инстанция на клас</li> <li>• Данни</li> </ul>
<b>2.3. Обекти</b>	<ul style="list-style-type: none"> <li>• Декларира обект от съществуващ клас</li> <li>• Дефинира понятието „конструктор“</li> <li>• Разбира предназначението на конструктор на клас</li> <li>• Създава обект от съществуващ клас.</li> <li>• Използва референция за достъп до елементи (членове) на текущия обект</li> <li>• Използва помощни средства на конкретна среда за програмиране</li> <li>• Използва подходящи коментари</li> <li>• Използва инструменти на средата за създаване на автоматична документация</li> </ul>	<ul style="list-style-type: none"> <li>• Конструктор</li> <li>• Референция за достъп до член на клас</li> <li>• Коментар</li> </ul>
<b>3. Членове на клас</b>		
<b>3.1. Член-променливи (свойства) на клас</b>	<ul style="list-style-type: none"> <li>• Разпознава членовете на класа, представени чрез UML диаграма на клас</li> <li>• Декларира свойства на клас</li> <li>• Описва синтаксис и семантика на методите за достъп до свойства и за промяна на данни</li> </ul>	<ul style="list-style-type: none"> <li>• UML диаграма на клас</li> <li>• Декларация на свойства в клас</li> </ul>



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"> <li>Генерира декларации на свойства със средствата на средата за програмиране</li> <li>Разбира предназначението и описанието на стандартни методи за достъп и промяна на стойности на член-променливи</li> </ul>	
<b>3.2. Методи на клас</b>	<ul style="list-style-type: none"> <li>Разбира предназначението на формалните параметри</li> <li>Подбира формални параметри на метод</li> <li>Извиква метод като съпоставя на списъка с формални параметри съответен списък с фактически параметри</li> <li>Различава технологията на предаване на параметри по стойност и чрез референция</li> <li>Разпознава връщащи и невръщащи стойност методи</li> <li>Описва метод, който не връща резултат</li> <li>Описва метод, връщащ резултат</li> <li>Задава коректно обръщение към метод</li> <li>Декларира локални променливи</li> <li>Посочва област на действие на променливи във фрагменти от програмен код</li> </ul>	<ul style="list-style-type: none"> <li>Параметри на метод</li> <li>Формални параметри</li> <li>Фактически параметри</li> <li>Параметри, предавани по стойност</li> <li>Параметри, предавани чрез референция</li> <li>Обръщение към метод</li> <li>Локални променливи</li> <li>Област на действие на променливите</li> </ul>
<b>3.3. Стандартен метод на клас</b>	<ul style="list-style-type: none"> <li>Използва стандартен метод на клас, връщащ форматиран текст с текущите стойности на данните на инстанцията</li> <li>Използва диалогови прозорци за въвеждане и извеждане на данни</li> </ul>	
<b>4. Създаване на графичен интерфейс</b>		
<b>4.1. Графични средства на интегрирана среда за създаване на графичен интерфейс</b>	<ul style="list-style-type: none"> <li>Използва интегрирана среда за програмиране за създаване на графичен интерфейс</li> <li>Посочва свойствата на елементи на графичен прозорец</li> <li>Създава елементи на графичен прозорец (етикет, текстово поле, бутон)</li> </ul>	<ul style="list-style-type: none"> <li>Модифицирана унгарска нотация за именуване</li> </ul>



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"> <li>• Задава настройки на свойствата на елементи на графичен прозорец</li> <li>• Описва добри практики за именуване на идентификаторите на графични елементи</li> <li>• Открива грешки при именуване на графични елементи</li> </ul>	
<b>4.2. Създаване на елементарен графичен интерфейс</b>	<ul style="list-style-type: none"> <li>• Реализира програмно достъп до и промяна на свойството „текст“ на графичните елементи</li> <li>• Реализира програмно обработката на събитие по подразбиране при натискане на бутон</li> </ul>	<ul style="list-style-type: none"> <li>• Обработка на събитие по подразбиране</li> </ul>
<b>5. Основни оператори</b>		
<b>5.1. Аритметични и логически оператори, релации за сравнение, оператор за присвояване и конкатенация</b>	<ul style="list-style-type: none"> <li>• Описва синтаксиса, семантиката и приоритета на основните оператори в конкретен език за програмиране</li> <li>• Разбира асоциативността на оператори с равен приоритет</li> <li>• Разбира спецификата на целочислената аритметика и аритметиката в смесени изрази.</li> <li>• Разбира семантиката на префиксен, инфиксен и постфиксен запис на оператор</li> <li>• Правилно прилага аритметични и логически операции, както и релации за сравнение в сложни изрази</li> <li>• Присвоява стойност на променлива</li> <li>• Разпознава случаи на неявно преобразуване на типове данни</li> <li>• Използва явно преобразуване на типове данни</li> <li>• Конкатенира текстови данни</li> </ul>	<ul style="list-style-type: none"> <li>• Унарни и бинарни оператори</li> <li>• Префиксен, инфиксен и постфиксен запис на оператор</li> </ul>
<b>5.2. Оценяване на логически изрази</b>	<ul style="list-style-type: none"> <li>• Разбира пълно оценяване на логически израз</li> <li>• Разбира частично (съкратено) оценяване на логически израз</li> </ul>	
<b>6. Конструкции за управление на програмата</b>		
<b>6.1. Алгоритми</b>	<ul style="list-style-type: none"> <li>• Описва основните видове изчислителни процеси – линеен, разклонен и</li> </ul>	<ul style="list-style-type: none"> <li>• UML диаграма на дейност</li> </ul>



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

	<p>цикличен</p> <ul style="list-style-type: none"> <li>Разпознава основните видове изчислителни процеси, описани с UML диаграма на дейност</li> <li>Обосновава необходимостта от прилагане на методи за избягване на дублиране на код</li> <li>Прилага подхода „отгоре-надолу“ (top-down) при решаване на задачи</li> </ul>	
<b>6.2. Линеини изчислителни процеси</b>	<ul style="list-style-type: none"> <li>Разбира последователността на изпълнение на линеен изчислителен процес</li> <li>Описва дизайн на линеен алгоритъм чрез UML диаграма на дейност</li> <li>Описва алгоритми, които се реализират чрез линеини изчислителни процеси</li> <li>Проследява изпълнението на линеен изчислителен процес</li> </ul>	<ul style="list-style-type: none"> <li>Линеен изчислителен процес</li> </ul>
<b>6.3. Разклонени изчислителни процеси</b>	<ul style="list-style-type: none"> <li>Разбира последователността на изпълнение на разклонен изчислителен процес</li> <li>Описва синтаксиса и семантиката на условен оператор „if“</li> <li>Разбира семантиката на вложени условни оператори</li> <li>Знае и може да прилага законите на Де Морган и закона за двойно отрицание над логически изрази</li> <li>Описва ефективни вложени условни оператори</li> <li>Описва синтаксиса и семантиката на оператор за множествен избор</li> <li>Реализира програмно условен преход със средствата на конкретен език за програмиране</li> <li>Конструира ефективни условни конструкции</li> <li>Проследява изпълнението на разклонен изчислителен процес</li> </ul>	<ul style="list-style-type: none"> <li>Разклонен изчислителен процес</li> </ul>
<b>6.4. Циклични изчислителни процеси</b>	<ul style="list-style-type: none"> <li>Описва синтаксиса и семантиката на оператори за цикъл в конкретен език за програмиране</li> <li>Избира подходяща циклична конструкция за решаване на конкретен проблем</li> <li>Може да моделира процес описан с една циклична конструкция чрез друг вид циклична конструкция</li> <li>Реализира програмно циклични процеси със средствата на конкретен език за</li> </ul>	



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

	програмиране <ul style="list-style-type: none"> <li>• Проследява изпълнението на цикличен изчислителен процес</li> <li>• Разбира семантиката на вложени циклични конструкции</li> </ul>	
<b>7.Капсулиране на данни и методи</b>		
<b>7.1. Капсулиране на данни и методи</b>	<ul style="list-style-type: none"> <li>• Анализира различни практики за моделиране на клас</li> <li>• Следва добри практики за моделиране на клас</li> <li>• Разбира необходимостта от скриване на данни и методи на клас</li> <li>• Описва необходимостта от прилагане на методи за достъп до данните в клас</li> <li>• Разбира разликата в представянето на променлива от числов и референтен тип данни</li> <li>• Разбира предназначението на конструкторите.</li> <li>• Обяснява предназначението на всеки от конструкторите – по подразбиране, за общо ползване, с параметри и за копиране</li> <li>• Описва конструктори на клас</li> <li>• Използва подходящ конструктор при създаване на обект</li> <li>• Разбира технологията презареждане на методи (<i>overload</i>)</li> <li>• Разбира приложението на технологията презареждане на методи при описание и обръщение към конструктори</li> </ul>	<ul style="list-style-type: none"> <li>• Модификатори за достъп „<i>public</i>“ и „<i>private</i>“</li> <li>• Числов тип данни</li> <li>• Референтен тип данни</li> <li>• Интерфейс на обект</li> <li>• Презареждане на метод (<i>overload</i>)</li> <li>• Конструктор за общо ползване</li> <li>• Конструктор по подразбиране</li> <li>• Конструктор с параметри</li> <li>• Конструктор за копиране</li> </ul>
<b>7.2. Видимост на променливите</b>	<ul style="list-style-type: none"> <li>• Описва областта на съществуване на променливите в даден клас</li> <li>• Разпознава локални за даден метод променливи</li> <li>• Описва областта на видимост на променливите в даден клас</li> <li>• Разпознава видимите в дадена област променливи</li> <li>• Генерира диаграма на клас с помощни визуални средства на интегрирана среда</li> <li>• От диаграма на клас генерира изходен код на клас с помощни средства на интегрирана среда</li> </ul>	<ul style="list-style-type: none"> <li>• Област на видимост на променливите</li> <li>• Припокриване на променливи</li> </ul>
<b>7.3. Статични членове</b>	<ul style="list-style-type: none"> <li>• Разбира предназначението на статичните членове на клас. Взема обосновано</li> </ul>	<ul style="list-style-type: none"> <li>• Статичен член на клас</li> </ul>



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

на клас	<ul style="list-style-type: none"> <li>решение за ползване на статичен член на клас</li> <li>Декларира, инициализира и използва в описанието на класа статични данни</li> <li>Дефинира статичен метод на клас</li> <li>Обяснява и прилага техниките за обръщение към статични членове на клас и данни на инстанция</li> <li>Разбира необходимостта от използване на константи</li> <li>Използва константи на стандартни класове</li> <li>Познава техники за дефиниране на константи</li> <li>Дефинира и използва потребителски константи</li> <li>Дефинира изброим тип данни</li> <li>Използва данни от изброим тип</li> </ul>	<ul style="list-style-type: none"> <li>Данни на инстанция</li> <li>Константа</li> <li>Изброим тип данни</li> </ul>
<b>8. Проектиране на графичен интерфейс</b>		
<b>8.1. Проектиране на графичен интерфейс</b>	<ul style="list-style-type: none"> <li>Проектира графична форма, представяща интерфейс на конкретно приложение</li> <li>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.</li> <li>Спазва добри практики за именуване на идентификаторите на графични компоненти</li> </ul>	<ul style="list-style-type: none"> <li>Компонента за избор в графична форма</li> <li>Многоредова текстова област</li> <li>Контейнер</li> </ul>
<b>9. Класове с потребителски типове данни</b>		
<b>9.1. Връзки между класове</b>	<ul style="list-style-type: none"> <li>Разбира основни връзки между класове – наследяване, генерализация, асоциация, агрегация, композиция</li> </ul>	<ul style="list-style-type: none"> <li>Наследяване</li> <li>Генерализация</li> <li>Асоциация</li> <li>Агрегация</li> <li>Композиция</li> </ul>
<b>9.2. Обектите като</b>	<ul style="list-style-type: none"> <li>Проектира клас, в който член-променливите са обекти от друг клас</li> </ul>	<ul style="list-style-type: none"> <li>Релация “има” (“has a”)</li> </ul>



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

членове на класове	<ul style="list-style-type: none"> <li>• Декларира член данни на клас, представляващи обекти от друг клас</li> <li>• Използва обекти на един клас в качеството им на член-променливи в други класове</li> <li>• Обяснява технологията на конструиране на обект, в който има данни от друг клас</li> <li>• Различава променливи и непроменливи референтни данни</li> <li>• Дефинира и използва изменяеми и неизменяеми референтни данни на клас</li> <li>• Конструира диаграми на класове, намиращи се в релация „има”</li> <li>• Реализира примери на релация „има” между класове при реализацията на графичния потребителски интерфейс</li> <li>• Познава особеностите на писането на свойствата на изменяеми референтни данни на обекти</li> <li>• Осигурява защита на данните чрез дефиниране на методи за достъп и контролирана промяна на референтни член-данни</li> </ul>	<ul style="list-style-type: none"> <li>• Неизменяеми референтни данни на клас</li> <li>• Изменяеми референтни данни на клас</li> </ul>
9.3. Модулна организация на класове	<ul style="list-style-type: none"> <li>• Разбира особеностите на вложени, локални и анонимни класове</li> <li>• Създава библиотека от потребителски дефинирани класове</li> <li>• Разбира предимствата на обединяване на класове в библиотека</li> </ul>	<ul style="list-style-type: none"> <li>• Видимост на класове</li> </ul>
10. Наследяване.		
10.1. Наследяване на класове	<ul style="list-style-type: none"> <li>• Разбира същността на релация „е” между два класа</li> <li>• Разбира технологията на наследяване на членове от базов клас при демонстриране на примери</li> <li>• Разбира технологията на наследяване в рамките на йерархия от наследственост</li> <li>• Познава основни методи на клас <i>Object</i> – проверка за „равни” обекти, представяне на обект във вид на символен низ</li> <li>• Реализира наследяване на базов клас</li> <li>• Използва наследени методи на базов клас</li> </ul>	<ul style="list-style-type: none"> <li>• Релация „е” (<i>is a</i>)</li> <li>• Базов клас</li> <li>• Наследник</li> </ul>
10.2. Капсулиране на данни при наследяване	<ul style="list-style-type: none"> <li>• Реализира потребителска йерархия на наследственост</li> <li>• Разбира правилата за онаследяване на член-променливи и методи от пряк и</li> </ul>	<ul style="list-style-type: none"> <li>• Йерархия на наследственост</li> <li>• Директен базов клас</li> </ul>



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"> <li>непък базов клас</li> <li>Спазва принципа за капсулиране на член-променливи при наследяване</li> <li>Разбира технологията на предефиниране на методи на базов клас</li> <li>Предефинира методи на базов клас</li> <li>Разбира принципа на конструиране на обект от наследен клас</li> <li>Спазва правилата при описание на конструктори на наследници</li> <li>Познава средство за достъп до данни и методи на директен базов клас</li> </ul>	<ul style="list-style-type: none"> <li>Индиректен базов клас</li> <li>Директен наследник</li> <li>Индиректен наследник</li> <li>Модификатор за достъп <i>protected</i></li> <li>Предефиниран метод</li> <li>Референция за достъп до методи и данни в директен базов клас</li> </ul>
<b>10.3. Полиморфизъм</b>	<ul style="list-style-type: none"> <li>Разбира същността на полиморфизма</li> <li>Разбира технологията на преобразуване „нагоре“ (<i>upcasting</i>) на произведен клас</li> <li>Прилага добри практики при преобразуване „надолу“ (<i>downcasting</i>)</li> <li>Реализира полиморфно поведение на референция към базов тип</li> <li>Познава синтаксиса и семантиката и използва оператор за проверка на съвместимост с типа на преобразуването „надолу“</li> </ul>	<ul style="list-style-type: none"> <li>Полиморфизъм</li> <li>Преобразуване нагоре (<i>upcasting</i>)</li> <li>Преобразуване надолу</li> </ul>
<b>10.4. Абстрактен клас</b>	<ul style="list-style-type: none"> <li>Обяснява ролята на абстрактния клас в обектно-ориентирания модел</li> <li>Разбира предназначението на абстрактните методи</li> <li>Декларира абстрактен метод в базов клас</li> <li>Реализира абстрактен метод на базов клас в неговите наследници</li> <li>Използва абстрактен клас за обработка на множество от обекти-инстанции на класове от йерархия на наследственост</li> <li>Използва свойствата на абстрактен клас при преобразуване „нагоре“ и „надолу“ в рамките на йерархия на наследственост</li> </ul>	<ul style="list-style-type: none"> <li>Виртуален метод</li> <li>Абстрактен клас</li> </ul>
<b>10.5. Интерфейс</b>	<ul style="list-style-type: none"> <li>Обяснява предназначението на чист абстрактен клас</li> <li>Обяснява предназначението на интерфейса в обектно-ориентирания модел</li> <li>Осъзнава ролята на интерфейса в обектно-ориентирания модел</li> <li>Посочва стандартни библиотеки за интерфейси</li> </ul>	<ul style="list-style-type: none"> <li>Интерфейс</li> <li>Реализация на интерфейс</li> </ul>



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"> <li>• Описва потребителски интерфейс</li> <li>• Придържа се към стандартен стил за именуване на интерфейс</li> <li>• Описва класове, реализиращи интерфейси</li> <li>• Обработва множество от инстанции на класове в йерархия на наследственост посредством интерфейс</li> <li>• Прилага интерфейс с преобразуване на тип „нагоре“ (<i>upcasting</i>) и „надолу“ (<i>downcasting</i>) в йерархия на наследственост</li> </ul>	
<b>11. Обработка на събития</b>		
<b>11.1. Обработка на събития</b>	<ul style="list-style-type: none"> <li>• Посочва основни събития с мишката</li> <li>• Описва обект на събитие на мишката</li> <li>• Асоциира събитие на мишката с функционалност на приложението</li> <li>• Посочва основни събития на клавиатурата</li> <li>• Описва обект на събитие на клавиатурата</li> <li>• Асоциира събитие на клавиатурата с функционалност на приложението</li> </ul>	<ul style="list-style-type: none"> <li>• Събитие</li> <li>• Обект на събитие</li> </ul>

## ХОРАРИУМ ЗА МОДУЛА

Годишен брой часове за изучаване на модула в 11. 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.



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	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	

## ОЦЕНЯВАНЕ

#### 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 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 базата на портфолио, съдържащо резултатите от практическа работа в клас и защитена домашна



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

работа. Желателно е оценката да отразява всички етапи от практическата разработка (анализ, модел, програмиране, 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%
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.



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

7.	Създаване 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 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.