

## ИМЕ НА КОНТРОЛА

Контролата, като обект на програмата, има име, чрез което е достъпна и може да се манипулира. Стандартните конвенции за именуване изискват имената на контролите да започват с префикс, указващ типа им.

Примерите по-долу показват основните контроли, името на компонентата в Java, стандартния префикс, който е желателно да се използва при именуването им, и тяхното предназначение.

### ПРИМЕРИ

Контейнер на контроли: Прозорец (форма)

Име по подразбиране: JFrame

Предназначение: Създава изпълнимо приложение с ГПИ и е основен контейнер.

Контейнер на контроли: Панел

Име по подразбиране: JPanel

Префикс: pnl

Предназначение: Може да се разполага във форма или в друг панел. Използва се и като основа за графика.

Контрола: Етикет

Име по подразбиране: JLabel

Префикс: lbl

Предназначение: Служи за визуализиране на едноредов текст. Текстът не може да се промени директно от потребителя.

Контрола: Текстово поле

Име по подразбиране: JTextField

Префикс: txt

Предназначение: Служи за въвеждане и излизане на едноредов текст.

Контрола: Бутон

Име по подразбиране: JButton

Префикс: btn

Предназначение: Служи за предизвикване на какво действие. Има текстов етикет.

Контрола: Стандартна диалогова кутия

Име по подразбиране: JOptionPane

Предназначение: Стандартна изскачаща диалогова кутия, която изисква въвеждане на текст или извежда информация.

## ПРАВИЛО

1. При проектиране на формата се уточняват броят, типът и разположението на графичните компоненти.
2. Типът се избира според предназначението на контролата.
3. При разположението се търси естетичен дизайн – подравняване, хармонични цветове, подходящи шрифтове и гр.
4. Бутонът за изход обичайно се разполага в голната гясна част на формата.
5. Едва след това се пристъпва към създаването на формата в среда за програмиране.

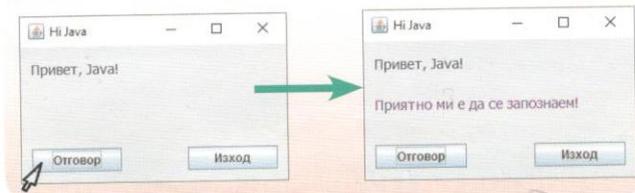
### Форма



## ЗАДАЧА

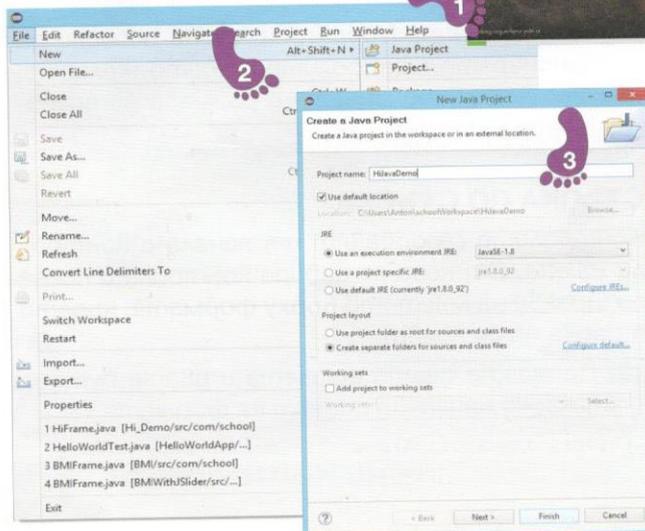
Създайте първото си приложение с ГПИ!

При натискане на бутона **Отговор** да се извежда текстът „Приветно ми е да се запознаем!“, а при натискане на **Изход** приложението да приключва работа.



## СТЪПКА ПО СТЪПКА – създаване на приложение

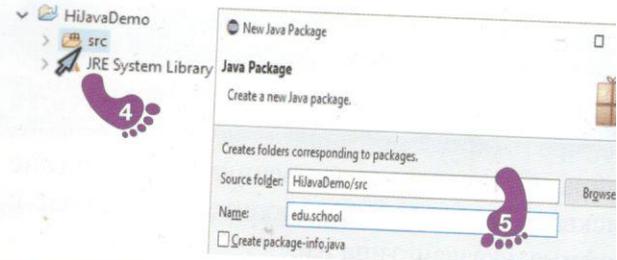
1. Стартирайте **Eclipse**.
2. Създайте нов проект от менюто **File** → **New** → **Java Project**.
3. Въведете име на проекта: **HiJavaDemo**.



## СТЪПКА ПО СТЪПКА – създаване на пакет

В панела *Package Explorer* вече имате папка на проекта!

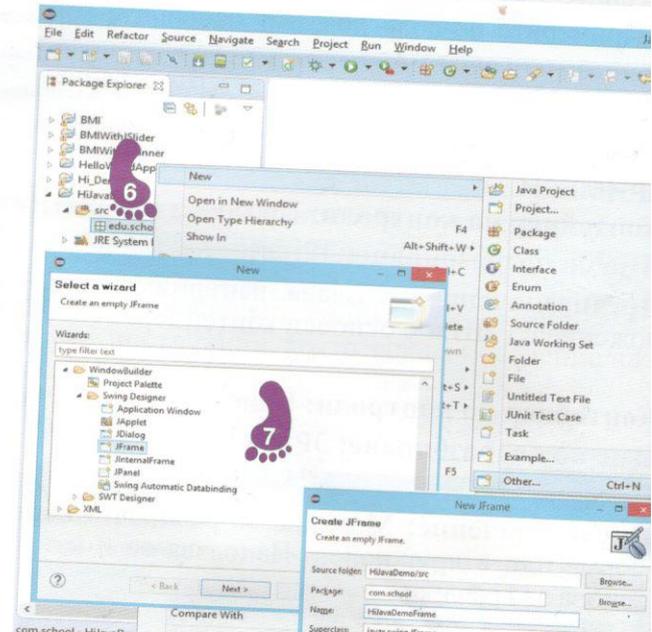
4. Създайте нов пакет, като щракнете с десния бутон на мишката върху папката **src** (вашият изходни кодове) и изберете **New** → **Package**.
5. Въведете име **edu.school**.



## СТЪПКА ПО СТЪПКА – създаване на форма

В папката с изходни кодове се появи пакет **edu.school**. В него ще *опаковаме* нашата нова форма.

6. Щракнете с десния бутон на мишката върху пакета в панела за преглед на структурата на приложението.
7. Изберете **New** → **Other** → **WindowsBuilder** → **JFrame** и натиснете бутона **Next**.
8. Въведете име на формата и натиснете **Finish**.
9. **Eclipse** създава програмен код, създаващ приложение с формата. Можете да видите как изглежда в графичния редактор – раздела *Design*.
10. Въпреки че няма други компоненти, приложението работи. Стартирайте го, за да се уверите!



## СВОЙСТВА НА ГРАФИЧНИ КОМПОНЕНТИ

Панелът *Свойства (Properties)* е контекстен – в него се настройват свойствата на контролата или контейнера, върху който сте щракнали.

## СТЪПКА ПО СТЪПКА

11. Задайте следните свойства за вашата форма:

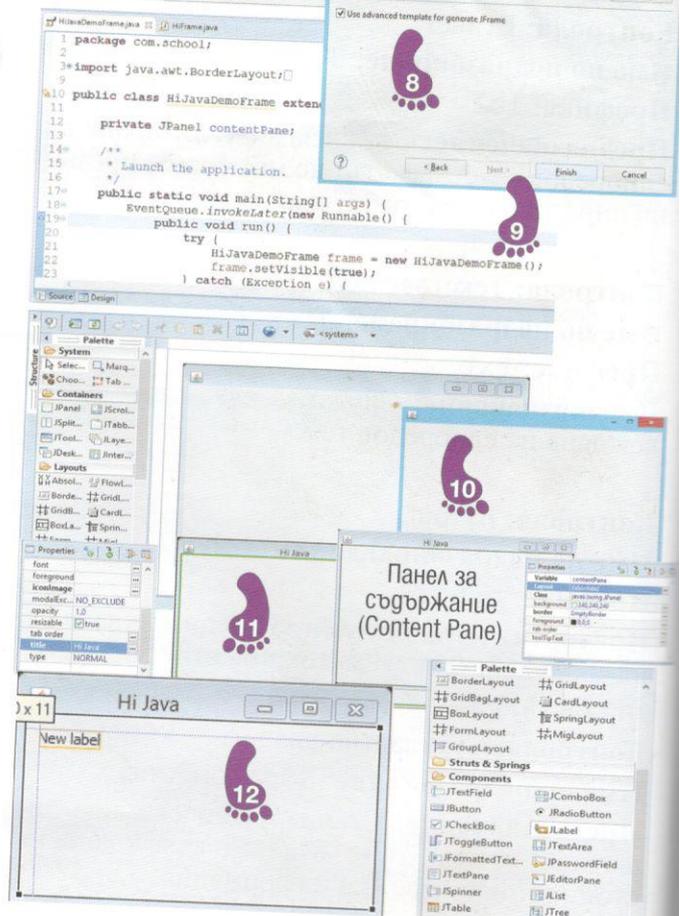
**Title:** *Hi, Java!*

**Layout:** *Absolute Layout*. Това ще ви позволи да позиционирате контролите, където желаете, независимо от позицията и размера на другите контроли.

## СТЪПКА ПО СТЪПКА

Контролите се добавят от панела с контроли *Palette*. От него се избира контрола и после се щраква на мястото върху формата, където трябва да се появи.

12. Добавете двата етикета и двата бутона съгласно началния проект на формата.



## ИМЕНУВАНЕ НА КОНТРОЛИ

Контролите се управляват от програмния код чрез техните имена. Правилното им именуване ви помага да се ориентирате за типа и предназначението им. Всяко име започва с префикс, свързан с типа на компонентата, и продължава с дума, описваща предназначението ѝ. Изписвайте втората дума след префикса, но с главна буква.

### ЗАМЕЧАНИЕ

При проектиране на формата всеки обект се именува веднага след разполагането му върху нея!

### СТЪПКА ПО СТЪПКА

12. Пренаменвайте всички контроли: щракнете върху контролата и след това променете свойството **Variable**.

Възможността за компоненти вече се виждат новите имена.

### ЗАДАЧА

Задайте цвят и стил на текста на всяка компонента. Използвайте свойството **text**, за да зададете надписите, които ще се виждат от потребителя.

Направете вторият етикет невидим, като зададете стойност **false** на свойството **visibility**.

## ФУНКЦИОНАЛНОСТ НА БУТОН

За да може приложението да реагира при щракване с мишката върху даден бутон, трябва да се опише **метод**, асоцииран със **събитието**, предизвикано от потребителя.

Методите са части от програмен код, чрез които се задава поведение на различните обекти. В тях може да се променят свойства на други компоненти или да се описват по-сложни серии от команди.

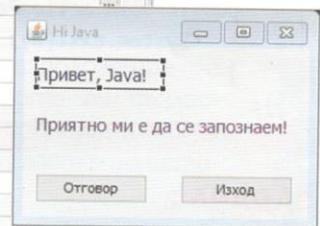
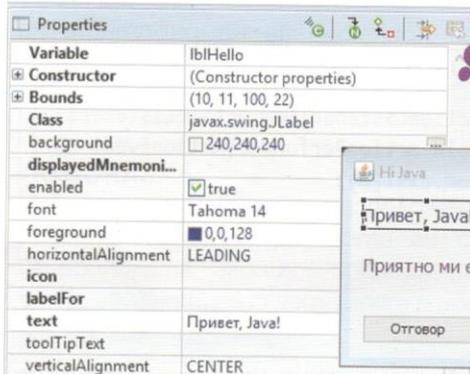
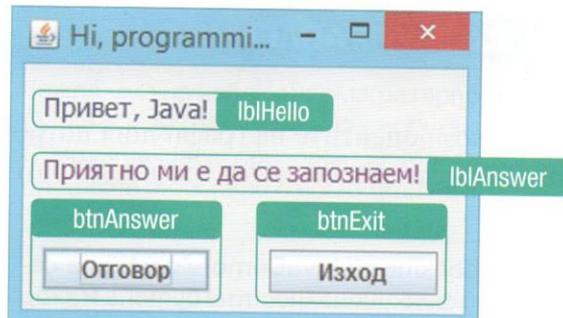
### СТЪПКА ПО СТЪПКА

14. Щракнете с десния бутон на мишката върху бутон **Отговор** (*btnAnswer*).

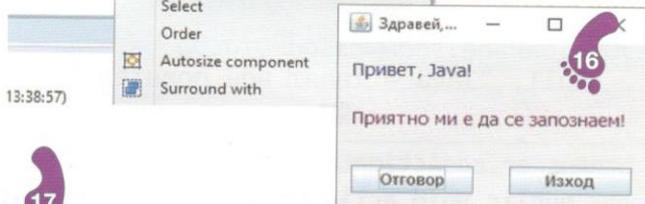
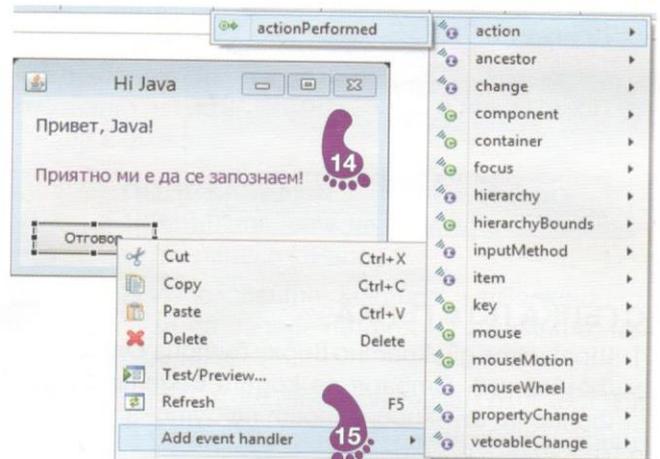
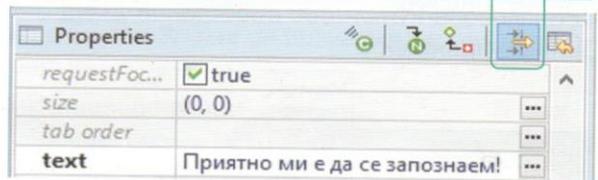
15. Изберете **Add event handler** → **Action** → **Action Performed**.

16. В създадения **метод** в програмния код запишете **lblAnswer.setVisible(true)**;

17. Стартирайте приложението и тествайте функционалността на бутон **Отговор**.



Ако не виждате някое свойство, използвайте бутона **Advanced Properties**.



```
btnAnswer.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        lblAnswer.setVisible(true);
    }
});
```

## 2.3.2. УПРАВЛЕНИЕ НА РАЗПОЛОЖЕНИЕТО НА КОМПОНЕНТИ

### В ТЕМАТА:

- ✓ Кой са компонентите на графичния потребителски интерфейс.
- ✓ Как умело да управляваме графичния потребителски интерфейс.

### ЗАДАЧА 1

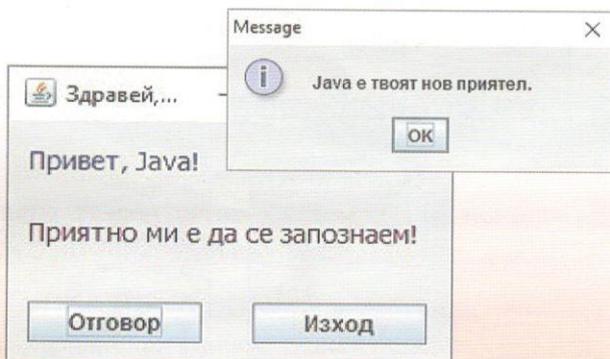
Отворете проекта **HiJavaDemo** и загайте самостоятелно функционалност на бутона **Изход** (*btnExit*).

Командата за приключване на работата на приложението е `System.exit(0)`;

```
btnExit.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        System.exit(0);  
    }  
});
```

### ЗАДАЧА 2

Добавете функционалност на бутон **Отговор** (*btnAnswer*) – при натискането му, освен визуализация на етикет *lblAnswer*, трябва да изскача и диалогова кутия със съобщение „Java е твоят нов приятел“.



### СТЪПКА ПО СТЪПКА

1. Щракнете двукратно върху бутона **Отговор** (*btnAnswer*). Отваря се кодът с метода, дефиниращ функционалност на бутона.
2. Добавете командата:

```
JOptionPane.showMessageDialog  
(null, "Java е твоят нов приятел.");
```

```
btnAnswer.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        lblAnswer.setVisible(true);  
        JOptionPane.showMessageDialog(null, "Java е твоят нов приятел.");  
    }  
});
```

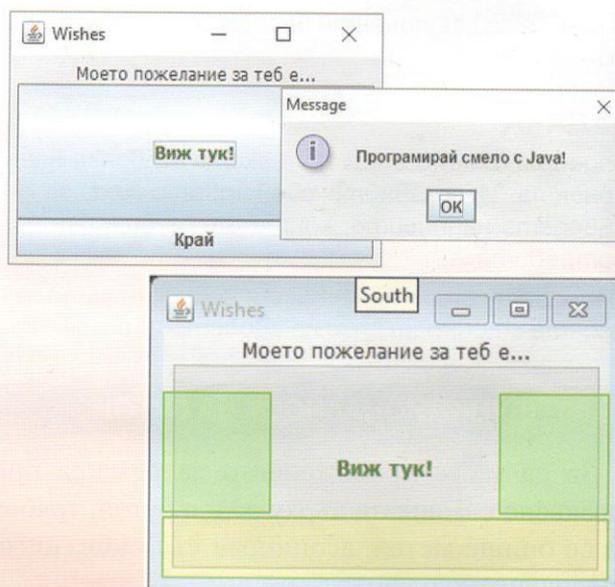
### ЗАДАЧА 3

Като следвате стъпките от предходния урок, създайте приложение **Wishes**, което при натискане на бутон да извежда в стандартна диалогова кутия пожелание към приятелите ви.

В стъпка 11 за стойност на свойството **Layout** на панела със съдържание, асоцииран с формата, изберете *BorderLayout*.

Наблюдавайте промяната на изгледа на формата в графичния редактор при добавяне на компонента.

Различава ли се изгледът на работещото приложение от този на предходното?

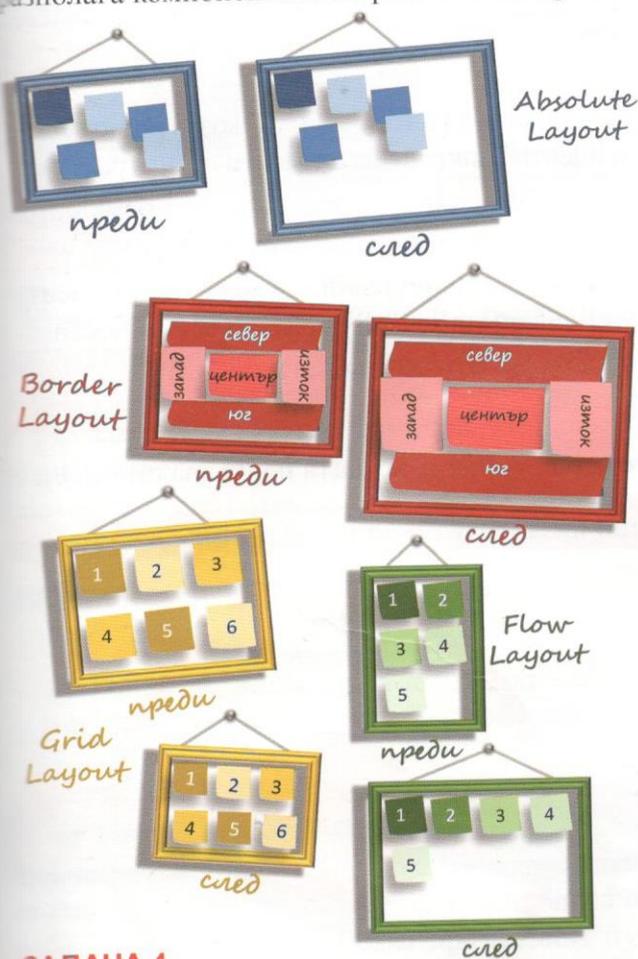


Имам нов приятел!



## УПРАВЛЕНИЕ НА РАЗПОЛОЖЕНИЕТО НА КОМПОНЕНТИТЕ

Инструментите за управление на разположението на графичните компоненти (**Layout Managers**) задават различни предварително дефинирани зависимости между взаимното местоположение на контролите и контейнерите. Изборът на подходящо разположение (**Layout**) е определящ за естетичния изглед и удобната навигация при преоразмеряване на прозореца или визуализация на различни монитори. **Absolute Layout** разполага компонентите на абсолютни позиции, зададени в пиксели, спрямо горния ляв ъгъл на панела за съдържание. **Border Layout** разполага компонентите елктно в пет региона – изток, запад, север, юг и център. **Flow Layout** разполага компонентите еднопосочно – в ред или колона. **Grid Layout** разполага компонентите в правоъгълна мрежа.

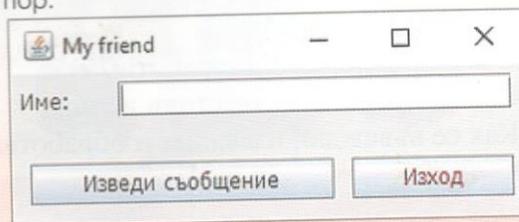


### ЗАДАЧА 4

Потърсете допълнителна информация за разположенията на компоненти в Java.

### ЗАДАЧА 5

Проектирайте форма, в която може да се въвежда име на човек. Формата трябва да има два бутона – за извеждане на съобщение и за изход. Създайте Java приложение **MyFriend**, в което опишете формата с помощта на графичния редактор.



#### ПОДСКАЗКА:

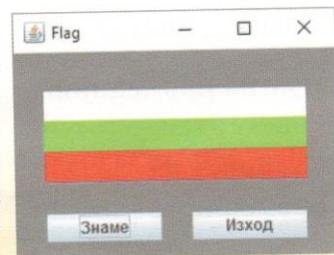
Изберете подходящо разположение за компонентите. Именувайте ги съгласно правилата. Задайте подходящи свойства, гарантиращи четимост и естетичен дизайн.



#### ПРЕДИЗВИКАТЕЛСТВО

Създайте приложение с ГПИ, което при натискане на бутон **Знаме** да визуализира българското знаме.

Модифицирайте приложението така, че да показва груп на национален флаг. Помолете съучениците си да го стартират и да познаят на коя държава е флагът.



#### ПОДСКАЗКА:

- 1) За цветовете на знамето използвайте панели (JPanel).
- 2) Използвайте метода `setBackground`, за да зададете фон на панелите, например: `pnlColor1.setBackground(Color.WHITE);`
- 3) Отстранете синтактичната грешка, като добавите в началото на кода декларация `import java.awt.Color;`



```
1 package edu.school;
2
3 import java.awt.Color;

btnDrawFlag.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        pnlColor1.setBackground(Color.WHITE);
        pnlColor2.setBackground(Color.GREEN);
        pnlColor3.setBackground(Color.RED);
    }
});
```