



# Среда за визуално програмиране

*Разработил: инж.А.Анчев*



# 1.Интегрирана среда

- ☛ Java е един от най-разпространените езици за програмиране. Популярността и ползването на Java се увеличава цели две десетилетия – това е голямо постижение, за който и да е програмен език.
- ☛ Има само няколко езика за програмиране, които не спират да растат през годините, и Java е един от тях.





# Защо да изучаваме JAVA?

- 1. **Обектно – ориентиран** - Java е обектно – ориентиран език, който поддържа всички принципи на обектно ориентираното програмиране – Абстракция на Данни, Енкапсулация, Полиморфизъм, Предефиниране и Наследяване. Това прави езика еквивалентен на C++.





- 2. **Богато API** - Друга съществена причина да научите Java е богатото **API (Application Programming Interface)** - пилотно-програмен интерфейс е интерфейсът на изходния код, който операционната система или нейните библиотеки от ниско ниво предлагат за поддръжката на заявките от приложния софтуер или компютърните програми.





- ☛ Java има API за почти всичко, което може да Ви потрябва, например: I/O, мрежово свързване, работа с XML, свързване с бази данни и т.н.
- ☛ Всичко останало се поддържа от Open Source библиотеки като Apache Commons, Google Guava, и други.





- 3. Огромна колекция от Open Source библиотеки - големи корпорации като Apache, Google и други са допринесли за създаването на огромен брой библиотеки за всякакъв вид задачи, което прави програмирането на Java по-лесно, бързо и евтино.





- 4. **Не зависи от платформа и е безплатен** - През деветдесетте това е била основната причина за популярността на Java. Идеята за платформена независимост и мотото на Java “**write once run anywhere**” е било достатъчно да привлече много разработчици към Java. И до днес това е причина Java да бъде толкова популярен програмен език.





- 5. Прекрасно Общество и Документация -  
Обществото от Java програмисти винаги е готово да помогне на начинаещи или напреднали. Дори експерти могат да ви помогнат безплатно по различни форуми за Java както и в StackOverflow. Javadoc е отлична документация и прави справките и ученето докато пишете код лесно.







## 2.Програма за отстраняване на грешки

- ❖ Дебъгване е самия процес на откриване на грешки с една програма (**debugger**) в друга, чрез специални програми които могат да изпълняват и проверяват в себе си продуктите на софтуерните инженери.
- ❖ Някои дебъгери могат да правят и по сложни неща, например да изпълняват програмата стъпка по стъпка (**Stepping**) или да я спрат в даден момент чрез **Break Point** (точка която се поставя на даден ред в програмния код с цел проследяване на стойностите на дадени променливи).





## 3. Проектиране на GUI

Характеристика	Описание
Прозорци (Windows)	Повече прозорци позволяват множествена информация да се представи на потребителския екран.
Икони (Icons)	Иконите могат да представят различна по тип информация. За някои системи иконите представляват файлове, за други – процеси.
Менюта (Menus)	Командите се избират посредством менюта вместо да се въвеждат на команден език.
Показване (Pointining)	Показващото устройство (като мишка) се използва за избор от меню или за маркиране на желани елементи от прозорец.
Графики (Graphics)	Графичните елементи могат да се комбинират с текст върху екрана.





## Предимствата на GUI са:

1. Относително лесен за изучаване и използване. Дори потребители с базови компютърни умения могат да се научат да използват интерфейса след кратко обучение.
2. Потребителят има много прозорци за взаимодействие. Превключването към нова задача е възможно без да се губи от погледа информацията, генерирана от предходната.
3. Бързина – работа върху целия екран позволява незабавен достъп до коя да е област или елемент.



# Принципи на проектиране

Принцип	Описание
Пригодност (User familiarity)	Интерфейсът трябва да използва термини и концепции, извлечени от опита на хората, които най-много използват системата
Цялостност (Consistency)	Интерфейсът би следвало да бъде цялостен по отношение на това, че подобните операции трябва да се активират по аналогичен начин
Минимална изненада (Minimal surprise)	Потребителите не трябва никога да остават изненадани от поведението на системата
Възстановимост (Recoverability)	Интерфейсът да включва механизми, които да позволяват на потребителите лесно да възстановят състоянието в нормален режим, след грешки
Направляване на потребителя (User guidance)	Интерфейсът би следвало да предоставя смислена обратна връзка при възникване на грешки и да предлага контекстно-зависими средства за помощ
Разнородност на потребителите (User diversity)	Желателно е да има подходящ набор от възможности за комуникация с различни категории потребители

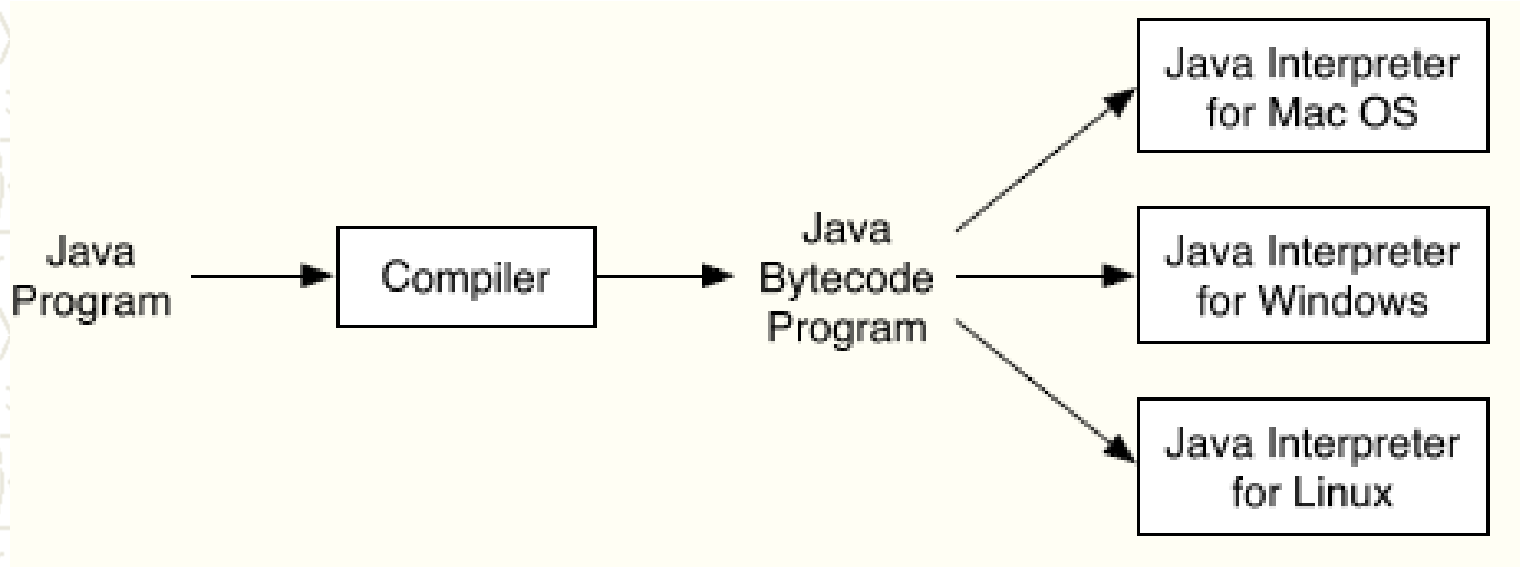




# Аплети

- Създателите на JAVA решават да използват комбинация на компилатор и интерпретатор. Програмите написани на JAVA се компилират на машинен език, но това не е машинен език за реалния компютър, а за т. нар. виртуален компютър наречен **виртуална машина на JAVA**.
- Машинният език за тази виртуална машина се нарича **байт-код за JAVA**. Основното предимство на този код е че може да се използва като машинен език за всеки реален компютър.







# Програмен код

```
Area.java x
Source History
1 package area;
2
3 /**
4  * Програма за определяне площ и обиколка на окръжност
5  * @author eng.Anchev
6  */
7 public class Area {
8
9     /**
10    * @param args the command line arguments
11    */
12    public static void main(String[] args) {
13        double radius, circum, area, PI=3.14159265359;
14        //---- Initialize variables
15        radius=1.;
16        circum=2.*PI*radius;
17        area=radius*radius*PI;
18        //---- screen output
19        System.out.println("Обиколка на окръжност = "+circum);
20        System.out.println("Радиус = "+radius +", Площ= "+area);
21    }
22 }
```





- ✦ Първият ред дефинира клас **Area**, който се съдържа в редовете между отварящата и затварящата скоба.
- ✦ Името на класа трябва да съвпада с името на файла където се запамятава кода.
- ✦ Всички програми се извършват в рамките на поне един клас. Думата **public** означава, че класът е достъпен за други програми.







- ☛ Следва основният метод (или функция) **main**, където се стартира програмата. Всяка JAVA програма, която се задейства от компютъра, а не от мрежата (браузър), трябва да съдържа този метод.
- ☛ Думата **public** означава, че методът е достъпен извън класа, което е необходимо за да може да бъде извикан метода от интерпретатора.





- ⚡ Думата **static** означава, че методът може да се извика без да се създаде обект на класа.
- ⚡ Думата **void** означава, че методът не връща стойност, а само изпълнява командите в тялото, между отварящата и затварящата скоба.

