**Задача 1. Посочете термина, който съответства на описанието:**

|  |  |
| --- | --- |
| Термин | Описание |
| camelCase | A naming convention to eliminate spaces in a name, but to ease readability with capitalization. |
| hierarchy | A group of related Java classes. |
| main method | The method inside a class that runs when the class is compiled and ran |
| code block | Sections of code that are enclosed inside a set of curly braces. {} |
| PascalCase | First letter uppercase and the first letter of each internal word capitalized. Example: SavingsAccount |
| constant | A named value that does not change |
| camelCase | First letter lowercase and the first letter of each internal word capitalized. Example: studentFirstName |
| import | A code statement in a Java class file that includes java code from another package or class. |
| keyword | A word that has a special function in the Java language, and cannot be used as names for classes, methods, or variables. |
| constructor | A special kind of method that is a template for an object. |
| arguments | Values that are sent into a method or constructor to be used in a calculation or substituted with values from the class. |
| access modifiers | Keywords used to specify the accessibility of a class (or type) and its members. Ex: public, private, protected, default |
| method | A block of code inside a class that is used to change or access information about the class. |
| primitive types | The group of Java data types that do not use the keyword new when declared or initialized. They store the value in the same place in memory as the variable name. |
| byte | The smallest java primitive type (1 byte) that can hold an integer value. |
| long | This data type (8 bytes) is the largest integer type. |
| convention | The formatting and naming standards that most programmers follow. |
| int | This Java primitive data type (4 bytes) can hold integer values. |
| double | This Java primitive data type (8 bytes) is the largest primitive that can hold a decimal value. |
| initialization | When a variable is assigned a value for the first time. |
| float | This Java primitive data type (4 bytes) can be initialized with a decimal number preceding letter f. |
| char | A java primitive data type (2 bytes) that can hold single character values. Example: “a”, “#”, or “X” |
| scope | Used to describe the block of code where a variable exists in a program. A block of code is denoted by {}. |
| short | A Java primitive data type (2 bytes) that holds integer numbers within a shorter range than an int. |
| boolean | A one-bit java primitive type that can hold the value true or false. |
| concatenation | Joining multiple String objects together. |
| escape sequence | Specific characters that are preceded by a \ character. When evaluated, the special character is evaluated as a special function, such as tabs, newlines, etc. |
| ternary operator | A shorthand form of an if/else statement. |
| Scanner | A Java class used for reading keyboard or file input during program execution. |
| if/else statement | A type of program control that allows different segments of code to execute when the input value matches a given condition. |
| do/while-loop | A post-test loop that executes an unknown number of times until a condition is met, but always executes the first time through the loop. |
| for-loop | A pre-test loop that uses an iterator to keep track of how many times a loop will execute. |
| continue | A keyword used to skip over the remaining code in a loop and return program control to the beginning of the loop to execute again. |
| while-loop | A pre-test loop that executes an unknown number of times until a condition is met. |
| break | A keyword used to terminate a loop from executing before the loop condition is met. |
| array | A structure that stores multiple values of the same data type |
| algorithm | A logical computational procedure that if correctly applied ensures the solution of a problem |
| 2-dimensional array | An array of arrays, similar to a table, matrix, or spreadsheet. |
| nested for-loop | A for loop inside of a for loop |
| catch | A keyword in Java that signals the following block of code handles a specified exception. |
| throw | This stops the interpreter from running the rest of the code until it finds a catch. |
| syntax error | An error that indicates an issue with coding format. |
| runtime error | An error that occurs while the program is running, also known as an exception. |
| logical error | An error that occurs as a result of incorrect programmer logic. |
| try/catch statement | A block of code that handles exceptions by dealing with the exception if it is thrown. |
| exceptions | Errors that occur during run-time and can be corrected or handled by your code. |
| class | A template used for making Java objects. |
| object | An instance of a class. |
| this | Reference An optional keyword used to access the members and methods of a class. |
| new | The operator used to create an instance of a class. |
| garbage collector | A built-in function of the Java VM that frees memory as objects are no longer needed or referenced. |
| mutator | A method that changes the state of an object. |
| accessor | A method that returns information about an object back to the calling program. |
| method | A procedure (changes the state of an object) or function (returns information about an object) that is encapsulated as part of a class. |
| instantiate | A verb used to describe the act of creating a class object using the keyword new. |
| initialization | The process of assigning a default value to a variable. |
| null | An object reference that has not been instantiated. |
| finalizer | An optional method that is called just before an object is removed by the garbage collector. |
| constructor | A special method used to create an instance of a class. |
| public | A type of access modifier. Permits access from anywhere. |
| constructor | Used to assign initial values to instance variables of a class. |
| private | A type of access modifier. Permits access only from inside the same class. |
| static method | Is a method that is available for use without first creating an instance of the class. |
| static variable | Any Java class-level variable that is declared with the static modifier. This means only one instance of the class variable can exist in the JVM regardless of the number of class instances. |
| static | Is a keyword that makes a variable, method, or inner class available without first creating an instance of a variable. |
| default | When there is no access modifier. Same access as public, except not visible to other packages. |

**Задача 2. Посочете описанието на термина:**

|  |  |
| --- | --- |
| Термин | Описание |
| Наследяване | Когато друг клас наследява полетатата и методдите на друг клас, става негов подклас. |
| Суперклас | Клас, който бива наследяван. |
| Подклас | Клас, който наследява друг клас. |
| Ключова дума final (в контекста на наследяването) | Константа за класа. |
| Ключова дума extends | Използва се, за да покаже, че един клас наследява друг. |
| Множествено наследяване | Когато няколко класа наследяват един. |
| Пакет | Папка, в която са класовете; те имат достъп един към друг. |
| Статично импортиране | Позволява ни да ползваме статичните полета и методи на импортиран клас без да посочваме всеки път името му. |
| Предефиниране на методи | Когато подклас има различен код в метод, наследен от суперкласа (те имат същото име и аргументи). |
| Ключова дума super | Вика метод или поле на супер класа. |
| Многослойно наследяване | Когато клас наследява клас, който също наследява клас и т.н. |
| Абстрактен клас | Клас, от който не могат да се създават обекти, може само да се наследява. |
| Абстрактен метод | Метод, който трябва да се предефинира в подкласа. |
| Анонимен клас | Клас без име. |

**Задача 3. Обяснете смисъла на твърденията:**

* Частни членове (private) на суперкласът не се наследяват:
  + Private прави членовете достъпни само в същия клас.
* Обектът на подклас се създава “на части”:
  + Когато супрекласът създава част от подкласа?
* Между предефинирането и презареждането на методи съществува принципна разлика:
  + Предефинираните методи имат едно и също име и аргументи, а презаредените методи имат едно и също име, но различни аргументи.
* Методите в Java имат едно важно свойство, което се проявява при наследяването, то се нарича виртуалност:
  + Методите могат да презаписват методите на наследения клас.
* На основата на абстрактен клас не може да бъдат създавани обекти (но може да бъде декларирана обектна променлива):
  + Абстрактния клас забранява създаването на обекти, той служи само за обединение на подкласовете, от които може да се създават обекти.