Методи на Транслация

Свързване.

Време на свързване.

доц. д-р Александър Пенев

Свързване

Свързване

Понятието **свързване** съпътства всички поддейности на дейността програмиране.

- Създаване на език за програмиране;
- Реализация на език за програмиране;
- Писане на код на програма (кодиране);
- Тестване на програма;
- Зареждане на програма;
- Изпълнение на програма;
- и др.

Свързване

В ЕП **Свързване** наричаме дейността по приписването на някаква характеристика на някакъв програмен обект.

Характеристиката се избира от някакво предварително известно множество от характеристики/свойства.

Под програмен обект разбираме име (идентификатори), а всичко друго което свързваме с това име е описание на операцията свързване.

[S≡Име/Идентификатор; ←P≡Описание на свързването]

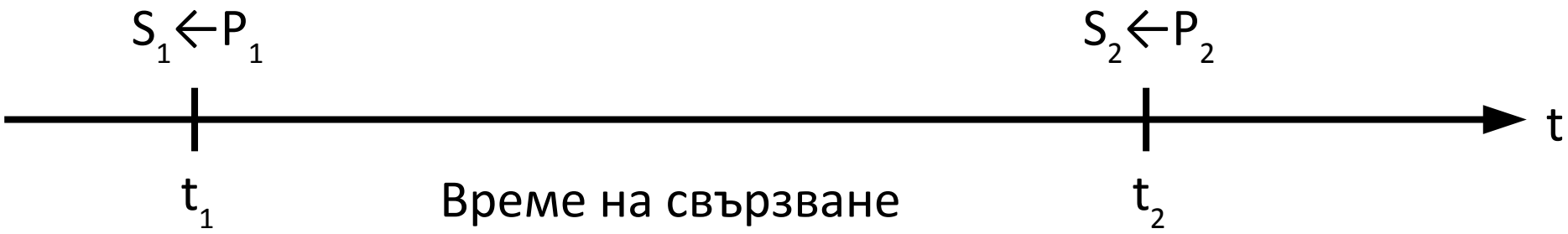
Свързване

Понятието **свързване** съпътства всички поддейности на дейността програмиране.

- Създаване на език за програмиране;
- Реализация на език за програмиране;
- Писане на код на програма (кодиране);
- Тестване на програма;
- Зареждане на програма;
- Изпълнение на програма;
- и др.

Време на Свързване

Време на свързване



- Свързване при дефиниране на ЕП;
- Свързване при реализация на ЕП;
- Свързване при „писане“ на програма;
- Свързване по време на транслация;
- Свързване по време на зареждане на програма за изпълнение;
- Свързване по време на изпълнение;

Свързване при Дефиниране на ЕП

Свързване при дефиниране на ЕП

- Всеки разработчик на ЕП фиксира състава и структурата на език, като определя различните типове множества от програмни обекти:
  - Елементи на езика;
  - Допустимите отношения между тях;
- Това означава че с обекта наричан ЕП се свързват някакви характеристики;
- Очевидно програмистът, който ще използва този ЕП не може да излезе извън рамките на възможностите / алтернативите / ограниченията, наложени от неговия разработчик;

Примерни обекти дефинирани при деф. на ЕП

Например лексиката и синтаксиса се дефинират по време на проектиране на ЕП и включват свързване на обектите от лексиката, синтаксиса и семантиката:

```
Идентификатор = буква | {буква | цифра}.
```

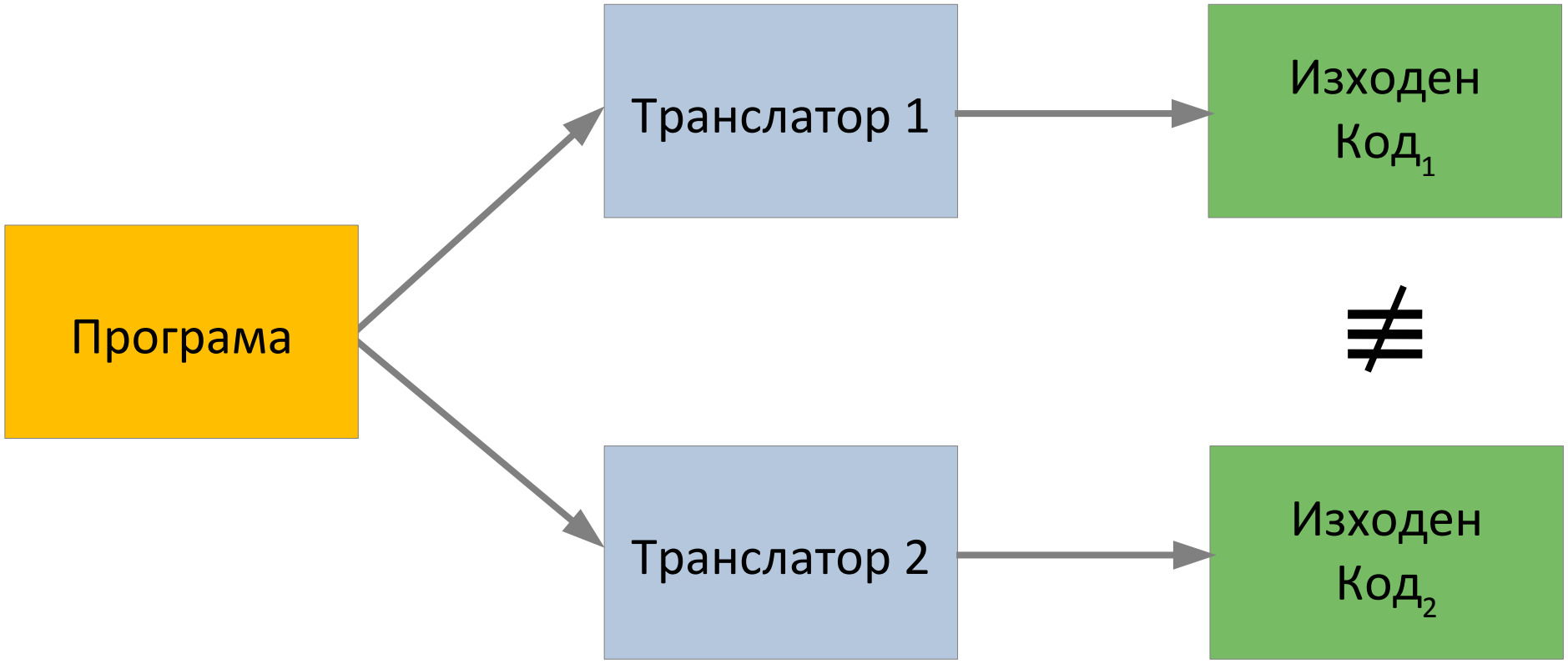
Това ограничава (дефинира) каква форма могат да имат идентификаторите в дадения език за програмиране т.е. свързва с понятието идентификатор възможните му обекти в езика.

Свързване при Реализация на ЕП

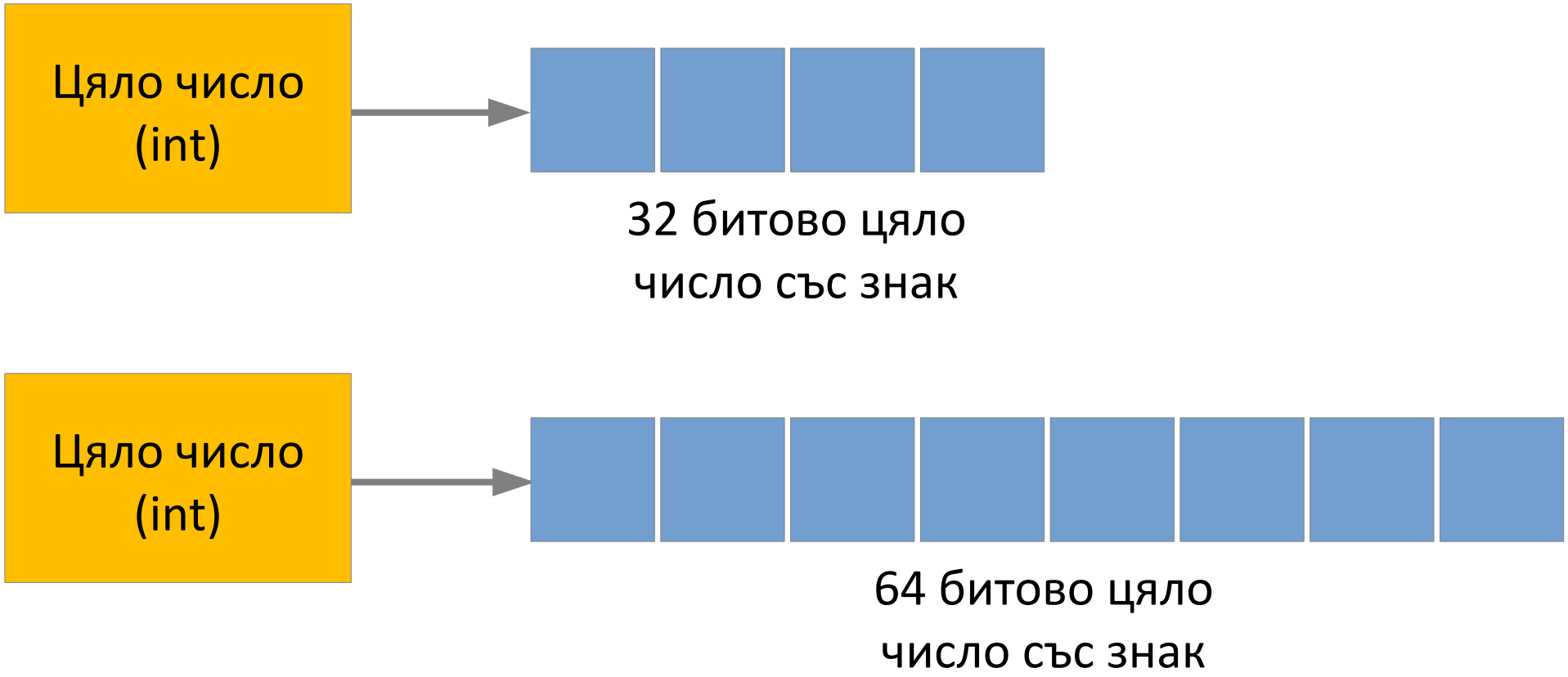
Свързване при реализация на ЕП

- Във всеки език има елементи, които не са напълно конкретизирани по време на неговото дефиниране. Например:
  - Представянето на числата и аритметичните операции върху тях;
  - Колко бита са целите/реалните числа
  - Какъв е алгоритъмът за изчисляване на корен квадратен и т.н;
- С тези елементи се свързват конкретни характеристики по време на реализацията на ЕП (може да зависи и от конкретната платформа);
- Това означава, че изходните програми, получени от различни реализации на даден ЕП могат да не бъдат еквивалентни програми. Въпреки това в повечето съвременни ЕП се правят опити за минимизиране на тази зависимост от реализацията;

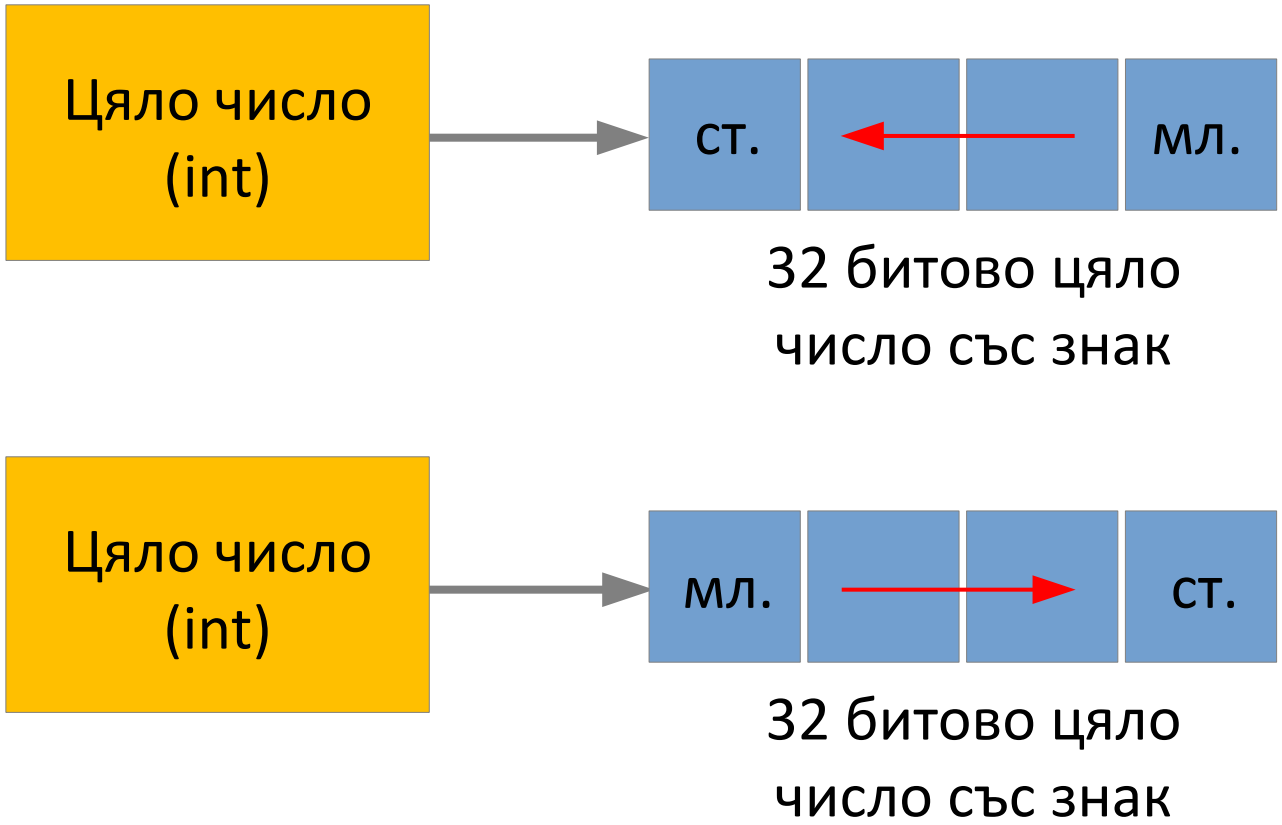
Пример


Програма
Транслатор 1
Транслатор 2
Изходен Код$_1$
Изходен Код$_2$
≠

Примерни обекти дефинирани при реализация



Примерни обекти дефинирани при реализация

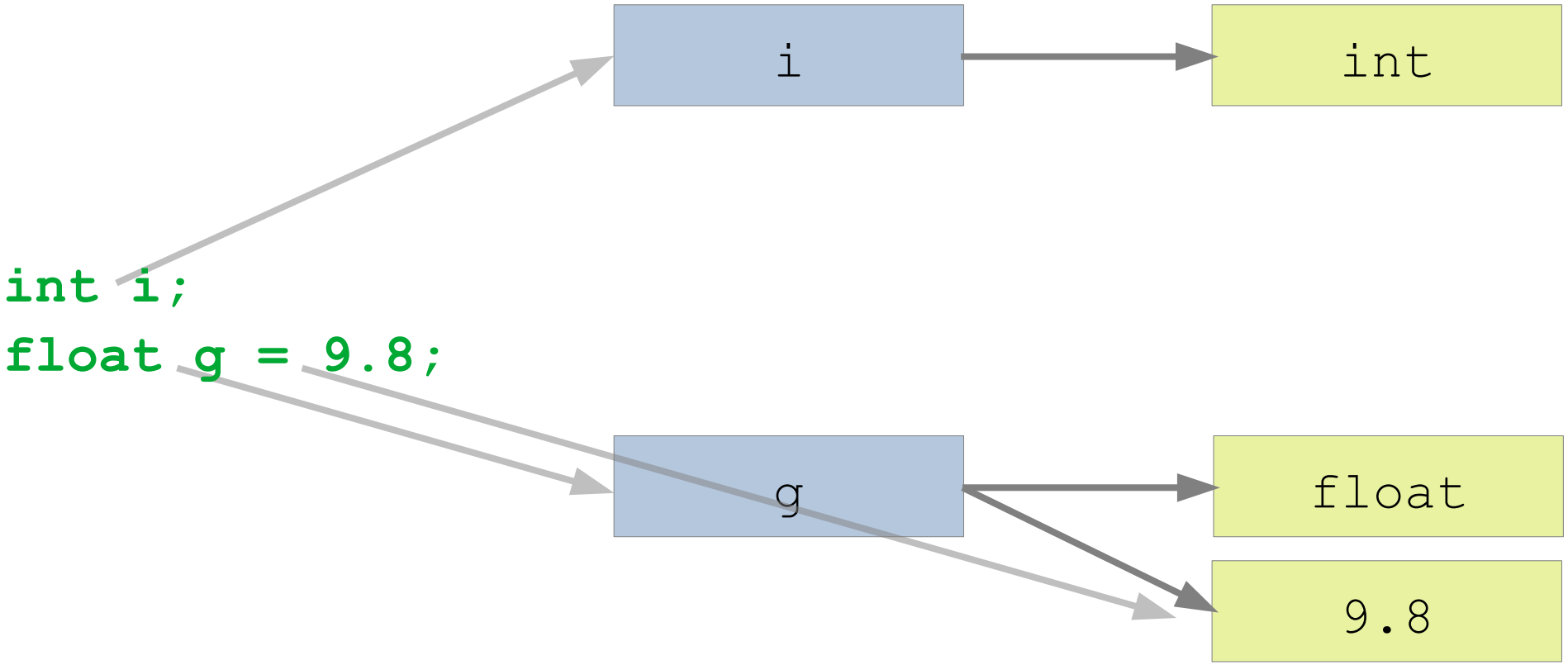


Свързване при „Писане“ на Програма

Свързване при „писане“ на програма

- Свързване при писане на програма е дейност при която:
  - Се дефинират програмни обекти;
  - Се дефинира последователност от промяна на характеристиките на така дефинираните програмни обекти;
- При писане на програма се използват възможностите на реализацията на ЕП. Например от множеството на всички възможни величини (<име, тип>) програмиста използва само няколко, като определя начина и последователността на промяна на текущата им стойност;

Пример дефиниране на променливи



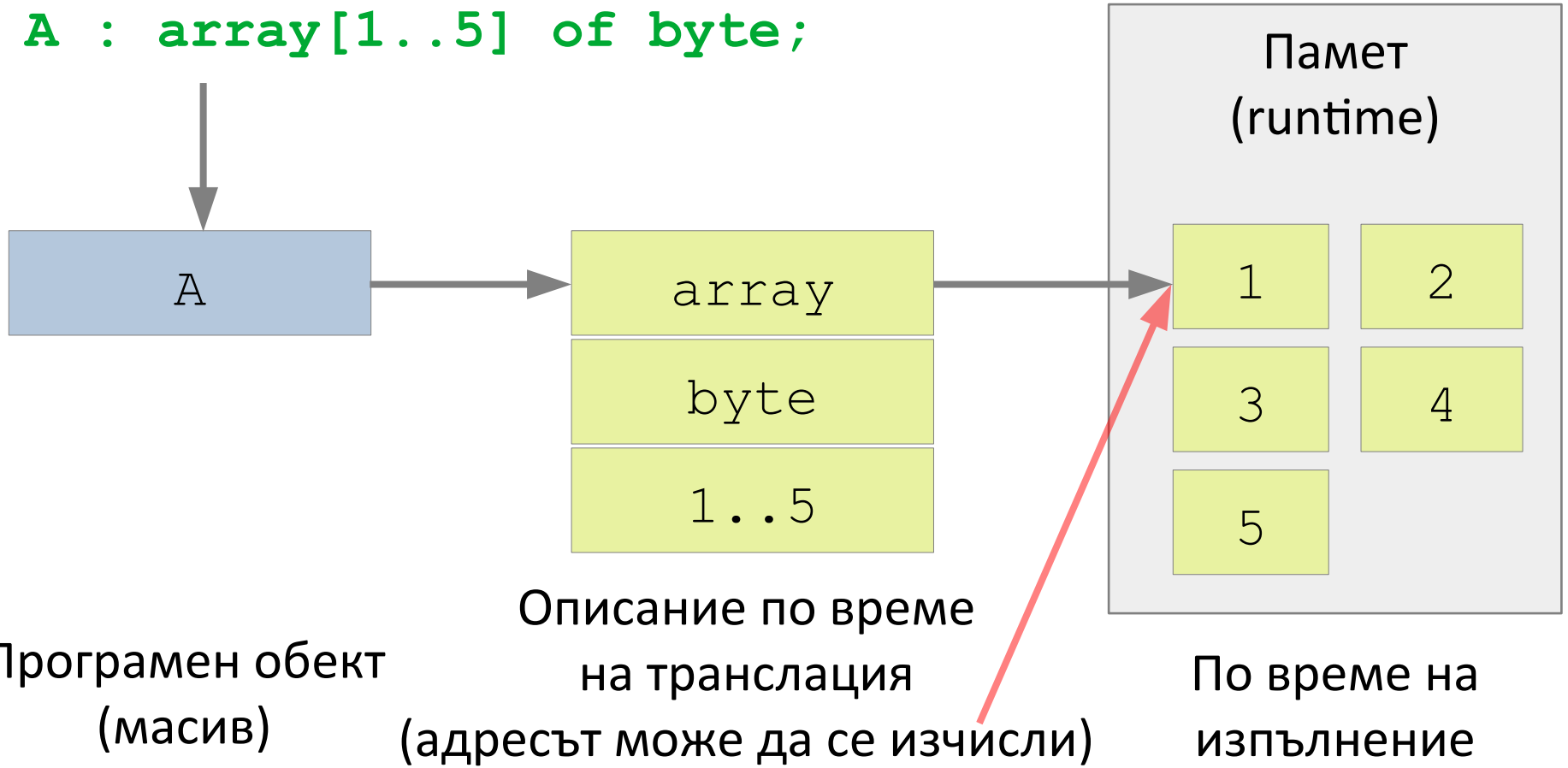
Свързване по Време на Транслация

Свързване по време на транслация

- Свързване което се извършва по време на транслация се нарича още **ранно свързване**. Пример за такова свързване е определянето на типа на величина или адреса и в ОП по време на транслация;
- Колкото по-голяма част от свързванията в една програма са ранни, толкова програмата е по-бърза, но е по-неефективна по отношение използвана памет;
- Ранното свързване налага и някои ограничения и неудобства за програмиста, например да пише само със статични променливи и методи;
- Повечето съвременни програми се състоят от отделни модули, които се компилират отделно. Това налага свързване на отделните модули (по време на транслация „статично свързване“ или по време на зареждане „динамично свързване“); (external)

Пример ранно свързване

```
var A : array[1..5] of byte;
```



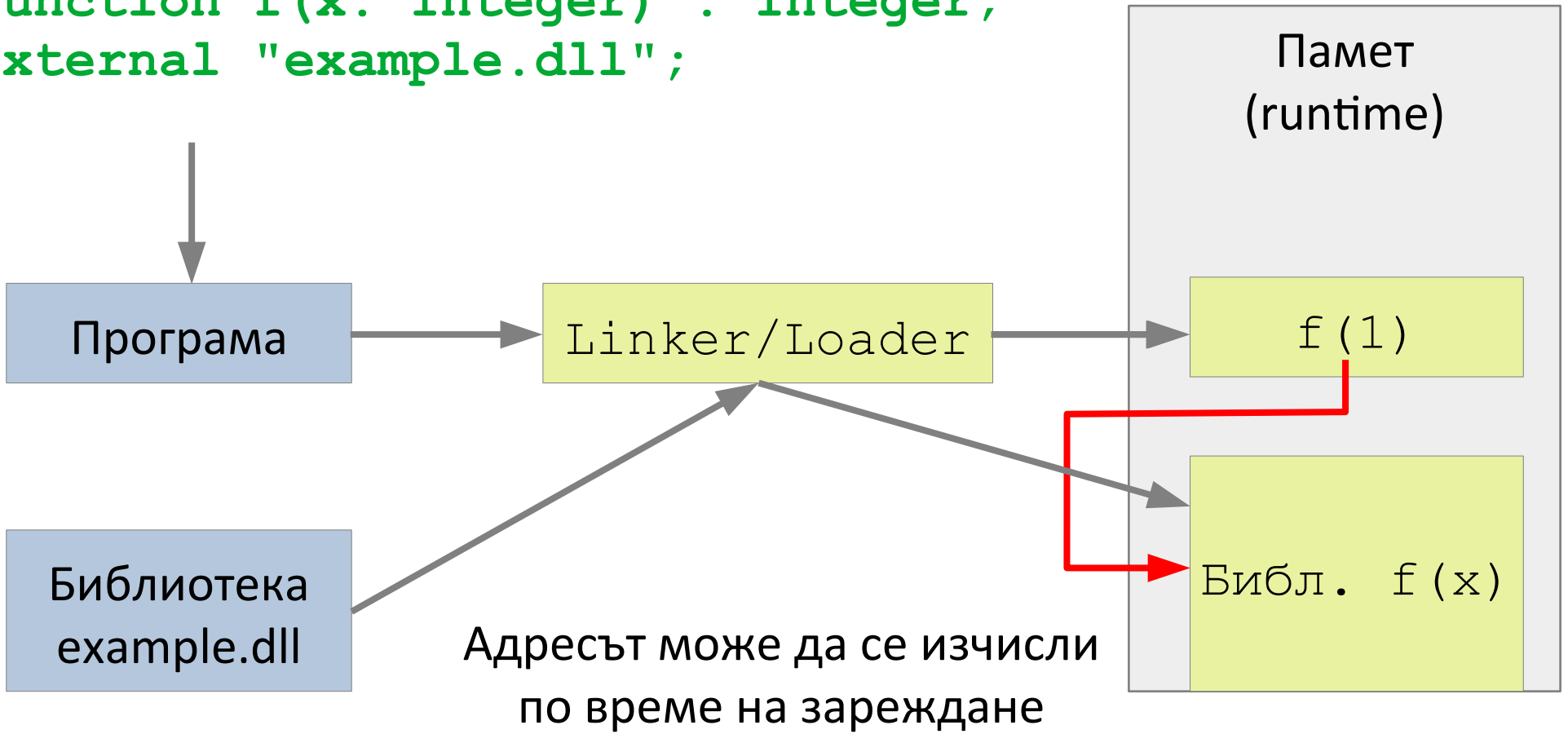
Свързване по Време на Зареждане на Програма за Изпълнение

Свързване по време на зареждане

- Свързването по време на зареждане е основно два вида:
  - Свързване с динамични библиотеки т.е. основната програма се зарежда заедно с други части компилиран код и някои техни обекти се свързват (функции, променливи и др.);
  - Свързване на изпълнимата програма със средата на ВИМ – настройване на преместваm код и др.;
- По своята същност свързването по време на зареждане е динамично и късно свързване (макар и са много ограничения в сравнение с късното свързване по време на изпълнение);
- Възможно е при различните стартирания на една програма да се заредят различни външни модули (трябва да се спазват ABI и API);
- Динамични библиотеки могат да се зареждат и в следващия етап – изпълнението на програмата;

Пример – свързване по време на зареждане

```
function f(x: Integer) : Integer;
external "example.dll";
```



Свързване по Време на Изпълнение

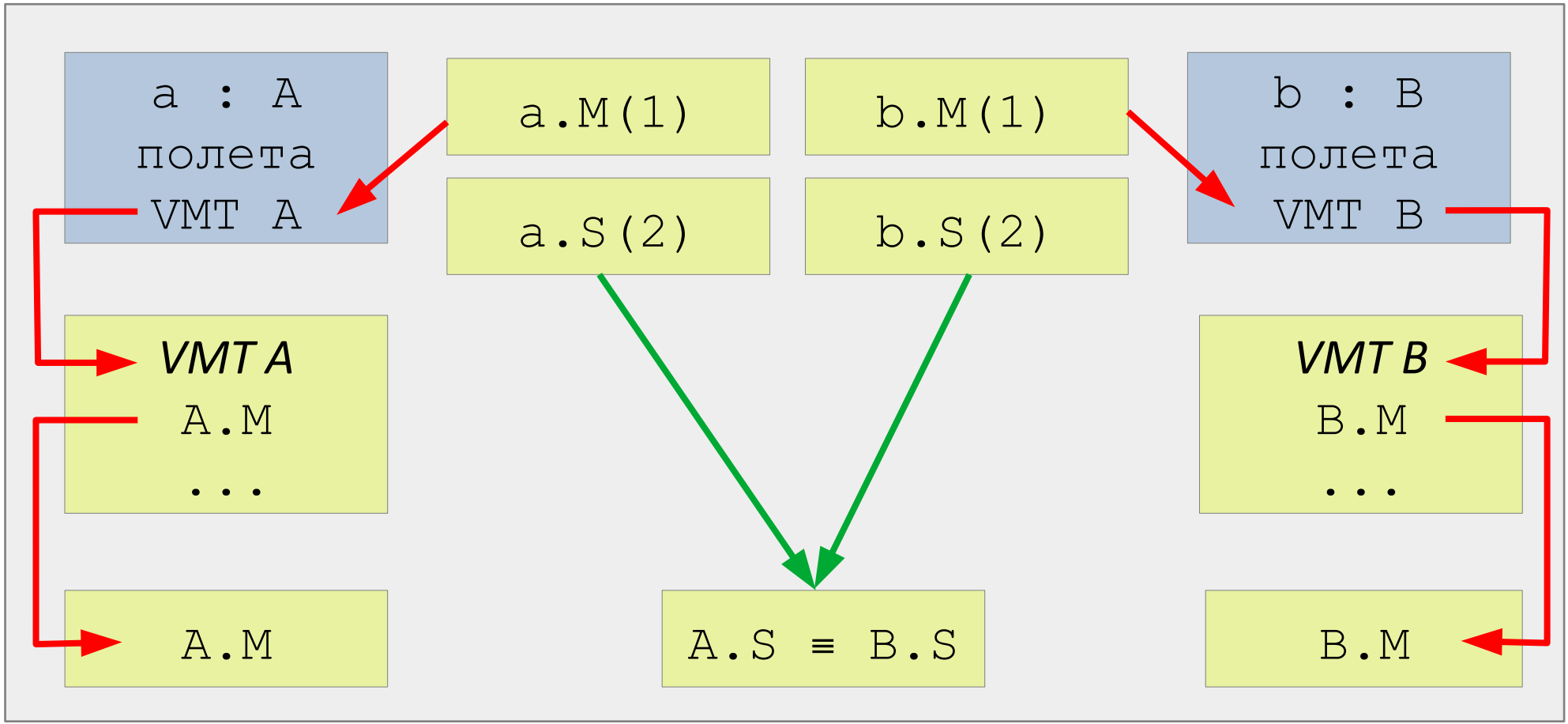
Свързване по време на изпълнение

- Това е така нареченото **късно свързване**. Например определяне на адреса в ОП на динамична променлива, адреса на виртуален метод, свързване на променлива с текущата и стойност и др.
- Има два важни подкласа късно свързване:
  - Свързване при вход/активиране на подпрограма;
  - Свързване в произволна точка от програмата;
- Очевидно колкото повече са късните свързвания в една програма толкова по-бавна е тя, но може да бъде направена ефективна по-отношение на използваната памет;

Пример – късно свързване (виртуален метод)

```
class A {
  public virtual void M(int x); {}
  Public void S(int x); {}
}
class B : A {
  public override void M(int x); {}
}
...
A a;
B b;
a.M(1); b.M(1);
a.S(2); b.S(2);
```

Пример – късно свързване (виртуален метод)



Изводи

Изводи

- При дефиниране на един ЕП, обикновено се вземат предвид и начина на реализация на допустимите в него свързвания т.е. този който дефинира език трябва принципно да е запознат с начина на реализация на ЕП;
- Замяната на ранно с късно свързване при реализация на ЕП не повишава гъвкавостта му значително, но влошава ефективността му;
- Незначително синтактични промени във версия на реализацията на ЕП твърде често означават промяна на времето на свързване. Например в Pascal:

```
var A : array[1..5] of byte;
```

е ранно свързване, докато

```
var A : array of byte;
```

е късно;

Въпроси?

apenev@uni-plovdiv.bg