

## Приложение Е. Диагностични съобщения

При използването на UniPascal е възможно да допуснете грешка при написването на програмата си. Грешките се делят на следните категории:

- предупреждения по време на компилация;
- грешки, откриваеми по време на компилация;
- грешки, откриваеми от операционната система по време на изпълнение на заявки на програмата (например опит за отваряне на несъществуващ файл);
- грешки, откриваеми по време на изпълнение, дължащи се на някакъв вид ограничение (например недостиг на памет);
- други грешки откриваеми по време на изпълнение;
- грешки, дефинирани от стандарта ISO и неоткриваеми от UniPascal;
- алгоритмични грешки.

Предупрежденията се дават само по-време на компилация. В някои случаи те са само предупреждение за възможна грешка, а в други за действителни грешки, които не влияят на компилацията на програмата (например използване на несъществуваща директива за управление на работата на компилатора).

Грешките, дефинирани от стандарта ISO и неоткрити от UniPascal обикновено не са грешки в UniPascal, поради някои разширения спрямо стандарта. Трябва да се има в предвид, че такива грешки могат да се появят, ако се опитате да използвате програмата в друга реализация на Pascal. Например не всички реализации на Pascal поддържат типа STRING.

Алгоритмичните грешки при никакви обстоятелства не могат да бъдат открити автоматично. В този случай програмата ви работи, но неправилно (т.е. не така, както искате Виe), а е възможно и изобщо да не работи (т.е. резултатът от работата ѝ да е непредсказуем).

### Е.1. Издавани от компилатора съобщенията за грешки

При откриване на грешка (или предупреждение) компилаторът записва в стандартния изходен файл номера на реда и самия ред, като под мястото на грешката поставя символа  $\wedge$ . На следващия ред ще бъде написано обяснение за вида на грешката, ако съществува файла **UNIPAS.ERR**, в който са записани помощни обяснения за всяка грешка. Ако такъв файл няма, извежда се само номера на грешката.

Грешките с номера от 1 до 13 са само предупредителни и независимо от тяхното наличие, компилаторът генерира изпълним файл. Грешките с номера от 240 до 255 са фатални и компилаторът не може да продължи работа след тях. При останалите грешки (с номера между 14 и 239) компилаторът продължава синтактичния анализ като вече не генерира изпълним файл, а само търси наличието на други грешки.

Каквато и програма да компилирате, не трябва да получавате грешка с номер по-малък от 0 или по-голям от 255, но ако това се случи, трябва да уведомите НИПЛ "Програмно Осигуряване", като запазите текста на програмата, в която тази грешка е настъпила.

#### Е.1.1. Предупредителни съобщения на компилатора

- 1: Директивата за управление на компилатора трябва да завършва до края на реда, въпреки че коментарите могат да бъдат на няколко реда.
- 2: Несъществуваща директива за управление на компилатора. Възможно е да сте сбъркали директивата. Например `{ $endif }` вместо `{ $endif }`.

- 3: Неправилно зададена директива. Например `{R*}` вместо `{R+}`
- 5: Използване на променлива от тип указател, която вероятно трябва да бъде последвана от `^` (но не е).
- 6: Използване на променлива от тип указател, която трябва да бъде последвана от `^` (но не е).
- 7: Базовият тип на указателя съвпада със себе си, т.е. указателният тип сочи към себе си.
- 8: Използвана е процедурата `FillChar` или `FillWord`, като зададената константна дължина е по-голяма от дължината на променливата.
- 10: Подаден е целочислен параметър на стандартната функция `ORD`. Това е излишно и обикновено е грешка. Например: `ORD(0)` вместо `ORD('0')`.
- 11: Не е установена дата и/или час в операционната система. Времето е важно при компилация на описателна част на модул (виж "Модули и техните версии").
- 12: Дължината на експортирания от асемблер идентификатор превишава 8 символа. Използват се само първите 8.
- 13: Неправилен (невалиден) символ в текста на програмата. Най-вероятно символът не е буква от стандартния ASCII набор, т.е. с ASCII код по-малък от \$20 или по-голям от \$7e (например коя да е буква от кирилицата). Компиляторът игнорира този символ, т.е. продължава да работи така, сякаш той не се е срещал изобщо.

### Е.1.2. Нефатални грешки

- 14: Компиляторът очаква, но не е намерил цифра.
- 15: Компиляторът очаква, но не е намерил число в интервала [0..255].
- 16: Компиляторът очаква, но не е намерил цяло число.
- 17: Очаква се ключовата дума 'program'.
- 18: Очаква се ключовата дума 'unit'.
- 19: Очаква се ключовата дума 'interface'.
- 20: Очаква се ключовата дума 'implementation'.
- 21: Очаква се ключовата дума 'begin'.
- 22: Очаква се ключовата дума 'do'.
- 23: Очаква се ключовата дума 'to' или 'downto'.
- 24: Очаква се ключовата дума 'end'.
- 25: Очаква се ключовата дума 'of'.
- 26: Очаква се ключовата дума 'then'.
- 27: Очаква се ключовата дума 'to'.
- 28: Очаква се ключовата дума 'until'.
- 29: Очаква се ключовата дума 'procedure' или 'function'.
- 30: Очаква се ключовата дума 'case'.
- 31: Очаква се ключовата дума 'for'.
- 32: Грешка в числена константа (реална или цяла).
- 33: Очаква се ключовата дума 'if'.
- 34: Очаква се ключовата дума 'while'.
- 35: Очаква се ключовата дума 'with'.
- 36: Компиляторът очаква, но не е намерил етикет (като идентификатор или число).
- 37: Очаква се ':' или ','.

- 38: Очаква се ключовата дума 'begin'.
- 39: Очаква се '='.
- 40: Очаква се ':='.
- 41: Очаква се '..'.
- 42: Очаква се '.'.
- 43: Очаква се '['.
- 44: Очаква се ']'.
- 45: Очаква се '('.
- 46: Очаква се ')'.
- 47: Очаква се ';'.
- 48: Очаква се ':'.
- 49: Очаква се ','.
- 50: Очаква се идентификатор
- 51: Грешка в раздела на декларациите.
- 52: Повторно деклариране на идентификатор.
- 53: Грешка в описанието на типа.
- 54: Грешка в прост тип (simple type).
- 55: Компиляторът очаква, но не е намерил целочислена константа за дължина на string.
- 56: Границите на диапазонния тип трябва да са от дискретен (ordinal) тип.
- 57: Долната граница надвишава горната граница.
- 58: Несъвместими по тип граници.
- 59: Недекларирана процедура или функция, която е описана като forward. Тази грешка се получава, ако сте описали forward процедура (функция) и след това не сте дали нормалното ѝ описание или ако в implementation частта не сте задали реализацията на процедура (функция), описана в interface частта на модула.
- 60: Неописан тип. Тази грешка се получава, ако не сте описали в същата type секция тип, който сте използвали като базов на указателен тип.
- 61: Този тип не може да бъде използван в качеството на индексен тип.
- 62: Не е разрешено използването на файлове на това място. Най-вероятно се опитвате да поставите файл като елемент на масив или запис, а това не е разрешено.
- 63: Предишната декларация не е forward. Тази грешка се дава само за процедури (функции) и означава, че вече има деклариран същия идентификатор.
- 64: Типът на функцията, зададен при изпреварващото описание (forward), не съвпада с този.
- 65: Резултатът на функцията може да бъде само от прост или указателен тип (в частност не се разрешават масиви, записи или низове).
- 66: При деклариране на функция не е даден типа на резултата.
- 67: Повторно деклариране на процедура (функция) в interface частта на модул.
- 68: Повторно предварително описание на една и съща процедура (функция).
- 69: Inline процедурите не могат да бъдат описвани като forward.
- 70: Inline процедурите не могат да бъдат овърлейни.
- 71: Заглавната част (header) на forward декларирана процедура или функция не съвпада с дадената при описанието ѝ.
- 72: Грешка в списъка от параметри.

- 73: Параметрите от тип файл могат да се предават само по адрес (т.е. трябва да бъдат var параметри).
- 74: Външни файлове (параметри в заглавието на програмата) са разрешени само за програмите, но не и за модулите.
- 75: Даденият параметър на програмата не е стандартен файл.
- 76: Типът на вариантите при вариантната част на записа (tag field) може да бъде само дискретен (ordinal).
- 77: Константата на вариантната част не съвпада с типа на вариантния описател.
- 78: Грешка в списъка от полетата на запис.
- 79: Базовият тип на множество трябва да бъде дискретен (ordinal).
- 80: Не са разрешени предварителни (forward) описания на процедури (функции) в interface частта на модул.
- 82: Не е указан файл за свързване с външна процедура. Това се получава, ако дадена процедура е описана като external, а в момента не е активирана директивата `{ $L xxx }`
- 83: Външната процедура (функция) не е намерена в указания чрез директивата `{ $L xxx }` файл.
- 84: Етикетите трябва да бъдат в граници от 0 до 9999.
- 85: Използване на резервираното име exit за име на етикет.
- 86: Повторна декларация на етикет.
- 87: Етикети не са разрешени в interface частта на модул.
- 88,89: Грешка във файла, даден в директивата `{ $L xxx }`. Не може да бъде прочетен или форматът му не отговаря на изискванията. Прекомпилирайте го.
- 90: Файлът, даден за свързване, импортира други асемблерски идентификатори. Преди да компилирате програмата, свържете този файл с всички други асемблерски файлове, които са необходими.
- 91: Външната процедура (функция) не е намерена в указания за свързване файл.
- 92: Външната процедура (функция) не е преместваема. Това се получава, ако сте асемблирали подпрограмата с директивата ORG или ако при свързването на асемблерските подпрограми не сте указали явно, че искате да получите отново OBJ файл.
- 93: Неочаквана поява на ключовата дума 'begin'. В `{ $M+ }` тя се поставя само в началото на процедура или функция.
- 94: Етикетът не е деклариран в текущия блок. В UniPascal не е разрешен преход (goto) от тялото на една процедура в друга.
- 95: Недефиниран, но деклариран и използван, етикет.
- 96: Недеклариран етикет. Използване на етикет, който не е деклариран.
- 97: Повторно дефиниран етикет.
- 98: Компилаторът очаква, но не е намерил етикет.
- 99: Неправилен етикет (трябва да е в граници от 0 до 9999).
- 100: Грешка в аритметичен (логически) израз (factor).
- 101: Грешка в използването на променливата.
- 102: Изразът трябва да бъде от тип Boolean.
- 103: Избиращият израз (selector) трябва да бъде от дискретен тип.
- 104: Повторно използване на константа в case оператора.
- 105: Типът на константата не съвпада с този на избиращия израз (Selector), даден в началото на case оператора.

- 106: Очаква се израз от дискретен тип.
- 107: Типовете на елементите на множество трябва да бъдат съвместими един с друг.
- 108: Долната граница надвишава горната граница.
- 109: Не са разрешени повече от един етикет на оператор. Ако е необходимо, поставете празния оператор между тях, т.е. вместо Lab1: Lab2: ...; напишете Lab1: ; Lab2: ...;.
- 110: Очаква се израз от тип Boolean.
- 111: Управляващата променлива на FOR цикъл трябва да бъде локална за текущата процедура.
- 112: Неправилен тип на управляващата променливата във FOR цикъл.
- 113: Неправилен тип на израз.
- 114: Типът на израза не съвпада с този на управляващата променлива на FOR цикъла.
- 115: Структурата не е байт пакетирана.
- 116: Изразът не е съвместим с типа на стоящата в лявата част на знака за присвояване променлива.
- 117: Не е разрешено предаване на управлението от завършващата част на процедура, функция, програма или модул към същинската му част.
- 118: Неправилен тип на променлива.
- 119: Управляващата променлива на FOR цикъл не може да бъде от тип LongInt.
- 120: Подпрограмите на асемблер (MC6800 native code) трябва да започват с \$0000.
- 121: Няма такъв вариант в този запис.
- 122: Дължината е неизвестна по време на компилация.
- 123: Типът е несъвместим с типа на вариантния описател (tag field).
- 124: Параметри от тип процедура (функция) не се поддържат в тази версия на UniPascal.
- 125: Пакетираните променливи не могат да бъдат давани като фактически параметри-променливи.
- 139: Типът на променливата не е указателен.
- 140: Типът на променливата не е масив.
- 141: Индексът трябва да бъде от дискретен (ordinal) тип.
- 142: Типът на индекса не съвпада с декларацията.
- 143: Типът на променливата не е запис.
- 144: Няма такова поле в този запис.
- 145: Изразът не е от тип множество.
- 146: Константата от тип низ е с различна дължина в сравнение с очакваната при използването ѝ в смесен израз с packed array of char.
- 147: Тази процедура не може да бъде използвана като фактически параметър на стандартната процедура EXIT. Като параметър могат да се използват само обикновените процедури.
- 149: В тази стандартна процедура или функция не е разрешено да се дават като параметри текстови файлове.
- 150: Присвояване на резултата на тази функция не може да се прави тук. Вероятно се опитвате да дадете резултата на функцията извън нейното тяло.
- 151: Процедурите нямат резултат.

- 152: За функциите трябва да се даде резултат. Опитвате се да използвате RETURN, без да сте задали резултата. Ако това е необходимо да се направи (т.е. задали сте го преди това чрез присвояване на стойност на името на функцията) напишете RETURN().
- 153: В тази стандартна процедура (функция) се очаква нетипизиран файл.
- 154: Очаква се име на файл (променлива от тип STRING).
- 155: Очаква се променлива от файлов тип.
- 156: Типът на дадения израз не съвпада с типа, който се очаква.
- 157: Грешка в типа на параметъра на стандартна процедура или функция.
- 158: Неправилен тип на операндите (спрямо операцията). Например логическа операция между променливи от тип char.
- 159: Несъвместимост между типовете на операндите.
- 160: За този тип е разрешено сравнение само за равенство и неравенство.
- 161: Броят на параметрите не съвпада с дадените в декларацията.
- 162: Фактическият параметър трябва да бъде променлива.
- 163: Несъвместимост с типа на формалния параметър.
- 164: Очаква се прост тип.
- 165: Несъвместимост с големината на формалния параметър.
- 166: Проверка за строго включване не е разрешена.
- 167: Не е разрешено присвояване.
- 168: Не е разрешено използването на константа.
- 169: Никъде в тялото на функцията не е указан резултатът ѝ.
- 170: Компилаторът очаква променлива от тип типизиран файл.
- 171: Компилаторът очаква израз от тип Integer.
- 172: Променливи от този тип не могат да бъдат прочитани или записвани. Вероятно използвате масив или запис в конструкциите READ, READLN, WRITE или WRITELN. Трябва да ги обработвате поелементно.
- 173: Компилаторът очаква променлива от тип текстов файл.
- 183: Неправилен израз.
- 184: Несъвместими или неправилен тип на операндите на бинарна операция.
- 185: Деление на нула в константен израз.
- 186: Неправилен тип на константа.
- 187: Константата е извън очаквания интервал.
- 190: Ограничение на реализацията. Диапазон на типа LongInt не е разрешен.
- 191: Ограничение на реализацията. Сравнение на константи не се прави от компилатора.
- 192: Ограничени на реализацията. Използване на тази стандартна процедура с константен параметър не се прави от компилатора.
- 193: Ограничение на реализацията. Израз, в който участвуват реални константи, не се пресмята от компилатора.
- 194: Големината на типа е недопустима. Обикновено се отнася за тип, който е по голям от 64 килобайта.
- 195: Базовият тип на множество не е подинтервал на [0..255]
- 196: Твърде много и/или твърде големи по-размер променливи. Теоретически размерът на локалните променливи не трябва да надвишава 32767 думи (65536 байта = 64 килобайта).
- 211: Идентификаторът не е от необходимия клас (вид). Тази грешка се получава, ако се опитвате да използвате процедура там, където може да

- има само променлива или ако се опитвате да използвате идентификатор на тип като име на процедура, както и в много други случаи на неправилно използване на съществуващ идентификатор.
- 212: Недеклариран идентификатор.
- 213: Идентификаторът е описан в поне два модула, които програмата използва. Необходимо е да се укаже от кой точно модул възнамерявате да използвате идентификатора.
- 214: Неочаквана лексема. Най-вероятно е изпусната ';' на горния ред (или по-точно на края на предишния оператор).
- 215: Тук не е позволено използването на константа.
- 216: Типът на променливата или израза е несъвместим с очаквания от компилатора тип.
- 217: Неправилно използване на възможността за промяна на типа (Type Cast). Обикновено се отнася за промяна на типа на променлива, чиято дължина се различава от дължината на типа.
- 218: Компиляторът очаква променлива.
- 219: Указаният идентификатор не е идентификатор на процедура.
- 220: Указаният идентификатор не е идентификатор на функция.
- 227: Константата е извън границите на очаквания интервал.
- 228: Константата от тип низ не завършва до края на реда.
- 229: Константата от тип низ е твърде дълга. Максимално разрешената дължина е 255.
- 230: Процедурата е твърде голяма. Разделете процедурата на няколко отделни процедури.
- 231: Твърде много или твърде дълги константи. Тази грешка се получава, ако общата дължина на всички константи надвишава 32 килобайта. Съветваме Ви да организирате някои от тях като файл, който да се прочете по време на изпълнение.
- 232: Твърде много процедури или функции. Общият брой на процедурите и функциите в един сегмент не трябва да превишава 255. Обявете някои от процедурите, които имат вложени такива, като сегментни или разбийте програмата на няколко модула.
- 233: Твърде много използвани модули. Слейте някои от тях в един модул.
- 234: Сегментът е твърде голям. Разбийте го на два сегмента или два модула.
- 235: Твърде много сегменти. Не може да имате повече от 255 сегмента (овърлейни процедури и модули) в програмата.
- 237: Компиляторът очаква, но не е намерил `{ $endif }`. Тази грешка се дава в края на файла, който се компилира, и означава, че е имало `{ $ifdef ... }`, `{ $ifndef ... }` или `{ $ifopt .. }` и не е намерен съответен `{ $endif }` за него.
- 238: Общата дължина на дефинираните символи за условна компилация е по голяма от допустимата.
- 239: Неправилно използване на директиви за условна компилация. Например срещнат е `{ $else }` или `{ $endif }` без да има съответстващ `{ $ifxxx ... }`.

### Е.1.3. Фатални грешки

При откриване на фатална грешка, компилаторът издава съобщение и прекратява компилацията.

Част от описаните тук грешки възникват при опит за отваряне на файл. Една от възможните причини е, че вече са отворени максимален брой файлове. Максималният

брой едновременно отворени файлове се задава от командната линия на UniDOS с помощта на командата HANDLES. По премълчаване те са десет. От зададения брой (или от 10 по премълчаване) трябва да се извади 5, това са системните файлове: input, output, message, auxiliary и printer (в операционната система имената им са съответно: StdIn, StdOut, StdErr, StdAux, StdPrn). При работа на компилатора е необходимо той да може да отвори едновременно поне 4 файла. Това са: файла, в който е записан кода на самия компилатор (тъй като той е програма с овърлейни подпрограми), този с изходния текст на програмата, която се компилира, един работен файл и файл, в който ще се генерира кодът. Ако има включване на файл (директива include), броят на файловете се увеличава с по един на всяко вложено включване, т.е. ако в текста на програмата има две директиви за включване, необходим е още един файл, но ако втората директива е вложена в първата, необходимо е да бъдат отворени още два файла. Най-добре е да поставите командата HANDLES 16 във файла AUTOEXEC.JOB. Така ще могат да бъдат отворени до 16 (реално за потребителя 11) файла едновременно. Този брой в повечето случаи е достатъчен.

Друга възможна причина компилаторът да не може да създаде и/или отвори за четене файл е неправилно указване на път в конфигурационния файл. Например:

```
/T d:\temp
/S d:\sym
/B d:\bdy
/P d:\pgm
/U d:\sym
```

а такива поддиректории и/или дискови устройства няма.

- 240: Неправилно написана директива за вмъкване (include) или свързване (link) на файл.
- 241: В началото на програмата (преди ключовата дума 'program') не може да има директива за вмъкване или свързване на файл.
- 242: Твърде много влягания на директивата за вмъкване. Разрешени са не повече от 4 влягания.
- 243: Незаговорен коментар. В края на файла, който се компилира, има отворен коментар.
- 244: Неправилен .SYM файл или библиотека. Вероятно е разрушен файлът. Най-добре прекомпилирайте модула, чието използване предизвиква тази грешка.
- 245: Конфликт при използване на модулите. Вероятно някой от модулите е бил прекомпилиран, а използващите го модули не са прекомпилирани.
- 246: Грешка (входно/изходна) при четене на .SYM файл. Най-вероятно файла не може да се прочете поради физическа грешка на диска. Прекомпилирайте описателната част (interface) на модула.
- 247: Не съществува .SYM файл за използвания модул. Компилирайте описателната част (interface) на модула.
- 248: Грешка при записване във .PGM (.BDY или .SYM) файла върху диска. Грешката може да се дължи на физически дефект върху диска или на липса на място върху него.
- 249: Грешка при опит за създаване на резултатния файл (.PGM, .BDY или .SYM). Едни от най-вероятните причини са неправилно указване на пътя, където трябва да бъде създаден файла, физическа повреда на диска или липса на място върху него.



- 250: Файлът, указан в директивата на компилатора за вмъкване или директивата за свързване, не може да бъде намерен. Не сте указали правилно поддиректорията или името на файла. Не забравяйте, че суфикс на файлът по подразбиране няма.
- 251: Опит за компилация на реализационна част на модул, който не трябва да има такава (interface only unit).
- 252: Неочакван край на вмъквания файл. Тялото на всяка процедура или функция трябва да се намира изцяло в един файл. Възможно е и да не е затворен някой коментар.
- 253: Грешка при четене на текста на програмата. Тази грешка обикновено се дължи на физическа грешка при четене от дискетата.
- 254: Прекалено дълъг ред. Един ред не може да превишава 128 символа.
- 255: Неочакван край на текста за компилация. Физическият край на текста на програмата (модула) е настъпил преди нейният логически край. Липсва точка след ключовата дума END или някъде в тялото на предишните процедури (но не по-напред от последните декларации) липсва END на съответния BEGIN.

Забележка: Грешка номер 243 (незатворен коментар) се появява, само ако файлът завършва с отворен коментар. Например:

```
queue:= queue^.next; (* обработване на следващия елемент
end { while };
end;
=== Bottom of File ===
```

незатворен коментар.

Ако след незатворен коментар се срещне затваряща коментарна скоба в символен низ или в друг затворен коментар със същия тип коментарни скоби, тази грешка няма да се появи. Например:

```
queue:= queue^.next; { обработване на следващия елемент
end { while };
end;
=== Bottom of File ===
```

незатворен коментар  
коментарът се затваря от тази коментарна скоба

На компилаторът не му е известно къде завършва коментарът. Най-вероятно ще възникнат друг вид синтактични грешки, като например тези с номера 252, 255 (както в по-горния пример) или ще се появят грешки, свързани с недеklarирани идентификатори. Но може и да не възникнат никакви грешки (ако се окаже, че са коментирани няколко оператора и с това не се нарушава синтаксиса и семантиката на езика). Ако получената грешка е непосредствено след затворен коментар и няма никаква причина за нейното наличие, проверете дали някъде по-нагоре в програмата има незатворен коментар.

## Е.2. Издавани по време на изпълнение съобщения за грешки

Освен грешките, открити по време на компилация на програмата, възможно е да има грешки, които се появяват едва по-време на изпълнение. Каквато и програма да изпълнявате, не трябва да получавате грешки с номера по големи от 240, но ако това се случи обърнете се към НИПЛ "Програмно Осигуряване", като запазете програмата, която е предизвикала такава грешка (както текста, така и кода ѝ), след като сте се убедили, че при повторна компилация и изпълнение се случва същата грешка на същото място (възможно е просто програмата ви да е неправилно прочетена от дискетата) и че

вашата програма не записва нещо там, където не трябва. Ако разрушите системни области (включително и кода на собствената си програма в паметта), по нататъшната работа на компютъра е непредсказуема. В такива случаи е възможна поява на всякакви грешки, странно поведение на компютъра и т.н. Този тип грешки са най-трудни за установяване и откриване на произхода им. Опитайте да включите проверката за граници  $\{R+\}$  и/или поставете тестов печат в програмата.

Следва списък на грешките, дадени от интерпретатора на Y код.

- 1: Не достига памет. По време на работа на програмата не достига памет за локалните променливи на активирана процедура/функция или няма достатъчно количество памет за да се обслужи заявка за такава от стандартните процедури NEW/GetMemWords
- 3: Деление на нула или вземане по модул нула.
- 4: Препълване на низ. Например на променлива от тип string[3] присвоявате стойност 'error'.
- 5: Опит да се построи множество с елементи извън интервала [0..255].
- 7: Неправилно индексирание в променлива от тип низ. Индексът трябва да е в интервала [1..255]
- 8: Даден израз излиза извън разрешената му долна граница. Това се получава при включена директива  $\{R+\}$ .
- 9: Даден израз излиза извън разрешената му горна граница. Това се получава при включена директива  $\{R+\}$ .
- 32: Грешка при използване на стандартната процедура DISPOSE или FreeMemWords. Опит да се освободи памет, която не е била заета чрез New или GetMemWords (например памет принадлежаща на операционната система или незаета памет).
- 65: Препълване при използване на реални числа (например  $1e20 * 1e20$ ).
- 66: Неправилна операция при реалните числа (например  $\ln(0)$ )
- 67: Деление на нула при реалните числа.
- 68: Целочислено препълване при превръщане от реално към цяло число (например  $\text{Round}(1.234e30)$ ).
- 208: Няма достатъчно памет за изпълнение на програмата. Наличната памет е недостатъчна, за да побере кода на програмата (и модулите които използва) и нейните глобални променливи (и тези на модулите, които използва).
- 209: Грешка при четене на програмата от дискета.
- 210: Няма генериран код за съответния модул. Най-вероятно е разрушен файлът върху дискетата. Прекомпилирайте програмата и модулите.
- 211: Овърлейният сегмент не е намерен на очакваното място. Най-вероятно сте сменили дискетата в устройството. Както вече беше отбелязано, при овърлейни програми (модули), файлът в който се намират овърлейните процедури не се затваря. Ако имате незатворен файл, не е разрешено да изваждате дискетата от дискетното устройство. Ако имате само едно устройство, използвайте възможността на UniDOS да разпознава две логически устройства с едно физическо.
- 212: Неправилна версия на модула. Тази грешка може да получите, ако сте правили изменения в описателната част (interface) на модула и не сте прекомпилирали реализационната му част.
- 214: Опит за излизане от/към несъществуваща процедура. Вероятно разрушавате системни области в компютъра.

- 216: Рекурсивна инициализация на модули. Например модул А използва модул В, който от своя страна използва модул А. Това не е разрешено. Поне един от двата модула трябва да бъде подчинен на (да не зависи от) другия.
- 217: Неправилен формат на библиотеката (SYSTEM.UPL). Най-вероятно се дължи на разрушена от неправилен запис структура на библиотеката. Запишете си отново библиотеката от дистрибутивната дискета.
- 218: Неправилен формат на .BDY файл. Важи забележката за грешка 217. Прекомпирайте разрушения модул.
- 219: При изпълнение на програма, използваща явно или неявно даден модул, не е намерена неговата implementation част (.BDY файл). Името на този модул е написано на реда, над който се появява съобщението. Използвайте BLPATH, за да укажете къде трябва да се търсят реализационните части на модулите или пълното име на библиотеката. Ако вече сте го направили, проверете дали необходимите .BDY файлове и/или библиотеката са записани там.

### Е.3. Списък на входно/изходните грешки

Входно/изходните грешки с код по-малък от 128 (\$80) съвпадат с грешките на операционната система UniDOS.

- 0 (\$00): No error
- 1 (\$01): Invalid function number
- 2 (\$02): Invalid drive
- 3 (\$03): Disk write protected
- 4 (\$04): Address failure
- 5 (\$05): Data failure
- 6 (\$06): General failure
- 7 (\$07): Invalid sector
- 8 (\$08): Reserved
- 9 (\$09): Invalid media type
- 10 (\$0a): FAT failure
- 11 (\$0b): Path not found
- 12 (\$0c): Reserved
- 13 (\$0d): Too many open files
- 14 (\$0e): Access denied
- 15 (\$0f): File already open
- 16 (\$10): Invalid file handle
- 17 (\$11): Disk full
- 18 (\$12): File lost in directory
- 19 (\$13): Invalid name
- 20 (\$14): Root directory full
- 21 (\$15): Directory exists
- 22 (\$16): Attempt to remove the current directory
- 23 (\$17): Directory not empty
- 24 (\$18): Bad LSEEK position
- 25 (\$19): Reserved
- 26 (\$1a): Not disk file
- 27 (\$1b): Too many drivers installed
- 28 (\$1c): Not same device

- 29 (\$1d): File exists
- 30 (\$1e): Printer timeout
- 31 (\$1f): Printer general failure
- 32 (\$20): Printer out of paper
- 33 (\$21): AUX: timeout
- 34 (\$22): AUX: parity
- 35 (\$23): AUX: overrun
- 36 (\$24): AUX: framing
- 64 (\$40): Invalid number of parameters
- 65 (\$41): Bad command or file name
- 66 (\$42): Insufficient memory
- 67 (\$43): Failure in .PGM file
- 68 (\$44): Language not available
- 69 (\$45): Cannot execute .CMD file
- 70 (\$46): File cannot be copied onto itself
- 71 (\$47): File opening failure
- 72 (\$48): File creation failure
- 73 (\$49): File not found
- 74 (\$4a): Invalid parameter

Входно/изходните грешки с номера от 128 (\$80) до 255 (\$ff) се издават от интерпретатора на Y код. Те са свързани със семантиката на UniPascal и обикновено не зависят от операционната система.

- 128 (\$80): Опит за използване на файл, който не е бил отворен изобщо. Например опит за използване на SEEK, READ или WRITE над деклариран, но неотворен файл.
- 129 (\$81): Опит за отваряне (open, reset, rewrite или append) на файл, който вече е отворен.
- 130 (\$82): Опит за запис във файл, който е отворен само за четене.
- 131 (\$83): Опит за четене от файл, който е отворен само за запис.
- 132 (\$84): Опит за четене или запис от/във файл, който не е отворен за съответната операция (BlockRead / BlockWrite).
- 133 (\$85): Опит за извършване на операция за нетипизиран файл над типизиран.
- 134 (\$86): Опит за изпълнение на операция, несъответстваща на вида на файла (типизиран - нетипизиран).
- 135 (\$87): Опит за използване на операция за текстов файл (readln/writeln) над нетекстов.
- 136 (\$88): Стандартната процедура BlockRead/BlockWrite е обработила по-малко байтове от указаните с параметъра.
- 137 (\$89): Опит за четене след края на файла.
- 143 (\$8F): Опит за повторно отваряне на файл с режим, несъответстващ на първото му отваряне.
- 144 (\$90): Грешка във формата на цяло или реално число.